

Sistema Robótico Colaborativo utilizando Restrições Virtuais

Fernando Manuel da Silva Ribeiro

Dissertação

Orientador: Prof. Doutor António Manuel Ferreira Mendes Lopes

Co-Orientador: Prof. Doutor Paulo Augusto Ferreira de Abreu



Mestrado Integrado em Engenharia Mecânica

Opção de Automação

Julho de 2010

© Fernando Manuel da Silva Ribeiro, 2010

Versão 3.0

Mens agitat molem

Resumo

O principal objectivo desta tese é desenvolver um protótipo de um sistema robótico colaborativo utilizando restrições virtuais executando controlo de admitância que permita a continuação da investigação na referida área.

A manipulação colaborativa surgiu em aplicações como a movimentação precisa de grandes cargas em operações de montagem, para permitir o controlo do movimento de um manipulador robótico directamente por um operador humano que assim mantém a propriocepção. As restrições virtuais vieram complementar o controlo livre nos sistemas colaborativos e permitiram que se realizassem tarefas extenuantes e de grande exactidão através da criação de regras ao movimento efectuado pelo operador. Em cirurgias, por exemplo, este tipo de sistemas permite guiar o movimento da mão até ao ponto desejado enquanto reduz o seu tremor.

Nesta dissertação será descrita a resposta comportamental adoptada para o sistema colaborativo e os controladores que asseguram a movimentação em funções auxiliares como a referenciação ou a movimentação autónoma. Propõe-se o desenvolvimento, simulação e implementação de controladores capazes de permitir o controlo sem esforço do elemento terminal de um robô cartesiano por parte de um operador humano. Propõe-se ainda a criação de algoritmos que instituem restrições virtuais ao movimento livre. Tarefas que podem beneficiar de um tal sistema são, por exemplo, a medicina da reabilitação, cirúrgica, indústrias de montagem e processamento.

Serão apresentados resultados tanto de testes em simulação como no robô real, e tiradas conclusões a respeito da qualidade das soluções implementadas.

Palavras-chave: Sistema robótico colaborativo, restrição virtual, controlo de admitância.

Abstract

The main objective of this thesis is to develop a prototype of a collaborative robotic system using virtual fixtures, through admittance control to allow the research in this area.

The collaborative manipulation appeared in applications such as the precise handling of large loads in assembly operations. This allows the direct control of the movement of a robotic manipulator by a human operator that keeps the proprioception. The virtual fixtures completed the free motion control in collaborative systems to carry out strenuous tasks with a great precision by setting up rules on the movement. In surgery, for example, these type of system supports the movement of the hand to the desired point while reducing their tremor.

In this dissertation it is described the behavioral response adopted in the collaborative system and the axis controllers to ensure the drive on the auxiliary functions such as referencing or the autonomous movement. It is proposed the development, simulation and implementation of axis controllers to allow the free motion of the manipulator led by a human. It is also proposed the creation of algorithms that preconize virtual fixtures over the free motion. Collaborative systems using virtual fixtures has potential benefits in diverse applications such as rehabilitation, surgery, assembly and processing industries.

Both, simulation and real-robot tests will be performed, and conclusions will be drawn about the quality of the implemented solutions.

Keywords: Collaborative robotic system, virtual fixture, admittance control.

Agradecimentos

Gostaria de agradecer aos meus pais e irmã por me inculpirem os valores que hoje definem a minha personalidade. Eles apoiaram-me, acreditaram em mim e lutaram para que eu pudesse ter um futuro melhor.

Aos meus orientadores Professor António Manuel Ferreira Mendes Lopes e Professor Paulo Augusto Ferreira de Abreu, pela oportunidade dos conselhos. A todos com quem privei na minha vida estudantil, do jardim-de-infância à universidade. Professores, colegas e conhecidos, pelos bons e maus momentos, pelas pequenas conquistas e as grandes derrotas, mas sobretudo por me ajudarem a compreender que o mundo não acaba nos livros!

Aos meus tios, Artur e Augusta, por nestes cinco anos, tantas e tantas vezes recriarem um pedacinho do meu distante lar. Não me esqueço da restante família, em especial do meu primo João por ser o irmão, não de sangue, mas de coração.

Por último queria recordar aquele que foi sempre o meu modelo, aquele está gravado no meu nome, o meu avô Manuel.

A todos o meu muito obrigado,
Fernando Manuel da Silva Ribeiro

ÍNDICE GERAL

ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABELAS	XVII
Capítulo 1	1
Introdução	1
1.1 Motivação.....	1
1.2 Sistemas Robóticos Colaborativos	2
1.3 Restrições Virtuais	3
1.4 Controlo dos sistemas colaborativos (impedância vs admitância)	7
1.5 Desenvolvimento dos Sistemas Robóticos Colaborativos	8
1.6 Objectivos da dissertação	14
1.7 Estrutura da dissertação.....	15
1.8 Conclusões	16
Capítulo 2	17
Descrição dos meios utilizados	17
2.1. Introdução	17
2.2. Robô cartesiano	17
2.3. O Matlab/Simulink.....	19
2.4. A arquitectura Host/Target PC.....	21
2.5. Desenho e concepção do sistema de interface robô/operador	22
2.6. Conclusões	25
Capítulo 3	27
Descrição da estratégia de controlo do robô cartesiano	27
3.1 Introdução	27
3.2 Máquina de estados - <i>Stateflow</i>	28
3.3 Modelação do sistema físico e síntese de controladores	32
3.3.1 Modelação do sistema físico	32
3.3.2 Síntese de controladores.....	36
3.3.3 Avaliação experimental do controlador I-P.....	45
3.3.4 Avaliação experimental do controlador de admitância	46
3.4 Conclusões	50
Capítulo 4	53
Descrição e implementação das restrições virtuais	53
4.1 Introdução	53
4.2 Restrição virtual como lei de controlo	54
4.3 Restrições virtuais para alvos e trajectórias	56

4.3.1	Definição da direcção de referência	57
4.3.2	Comportamento do sistema perto do ponto alvo	58
4.4	Restrições virtuais para volumes	59
4.4.1	Tubo.....	59
4.5.1	Cone.....	60
4.5	Implementação das restrições virtuais	64
4.5.1	Ponto Alvo.....	66
4.5.2	Curva	70
4.5.3	Tubo.....	77
4.5.4	Cone.....	82
4.6	Conclusões.....	84
Capítulo 5	85
	Análise experimental das restrições virtuais	85
5.1	Introdução.....	85
5.2	Ponto Alvo.....	86
5.3	Curva	88
5.4	Tubo.....	89
5.5	Cone.....	92
5.6	Análise da coordenação visual/motora	94
5.7	Conclusões.....	96
Capítulo 6	99
	Conclusões e desenvolvimentos futuros	99
6.1	Conclusões.....	99
6.2	Desenvolvimentos futuros	101
Referências	103

ÍNDICE DE FIGURAS

Figura 1.1 - Exemplo de uma restrição virtual aplicada ao seguimento de trajectórias.	4
Figura 1.2 - Controlo em impedância.....	7
Figura 1.3 - Controlo em admitância.	8
Figura 1.4 - <i>Hardiman</i> construído pela <i>General Electric</i> (Szondy).	9
Figura 1.5 - Dispositivo PADyC (Laboratory for Intelligent Mechanical Systems, 2004).	10
Figura 1.6 - Dispositivo P-TER (Intelligent Machine Dynamics Laboratory).	11
Figura 1.7 – Dispositivo COBOT monociclo (Intelligent Machine Dynamics Laboratory).	12
Figura 1.8 - Dispositivo COBOT triciclo "scooter" (Laboratory for Intelligent Mechanical Systems, 2004).	12
Figura 1.9 - Exemplo da utilização de IAD's na montagem de componentes na indústria automóvel (Peshkin & Colgate, 1999).	13
Figura 1.10 - Dispositivo <i>Lokomat</i> ® (Hocoma).	13
Figura 1.11 - Uso do <i>ACROBOT</i> em cirurgia ortopédica (Davies <i>et al.</i> , 2004).	14
Figura 2.1 - Robô cartesiano utilizado neste trabalho.	18
Figura 2.2 - Arquitectura <i>HOST/TARGET PC</i>	21
Figura 2.3 - Nova interface do robô.	23
Figura 2.4 - Interface entre o transdutor de força e o sistema porta-ferramentas.	23
Figura 2.5 - Adaptador para os diversos punhos.	24
Figura 2.6 - Suporte para o punho.	24
Figura 2.7 - Carcaça ergonómica para integração no punho.	25
Figura 2.8 - Uso do punho.	25
Figura 3.1 - Diagrama da <i>Máquina de Estados</i> do sistema robótico.	29
Figura 3.2 - Diagrama do <i>super-estado: Zero da Máquina</i>	31
Figura 3.3 - Modelo de um sistema de transmissão com acoplamento de rigidez e amortecimento próprio.	32
Figura 3.4 - Modelo em <i>Simulink</i> de um eixo linear.	34
Figura 3.5 - Modelo simplificado em <i>Simulink</i> de um eixo linear.	35
Figura 3.6 - Diagrama de blocos do controlador I-P de velocidade.	38
Figura 3.7 - Modelo em <i>Simulink</i> do controlador I-P aplicado ao modelo simplificado do eixo linear.	38
Figura 3.8 - Simulação da resposta do sistema ao degrau utilizando o modelo simplificado do eixo.	40
Figura 3.9 - Diagrama de blocos do controlador I-P de velocidade com malha anti-windup.	41
Figura 3.10 - Diagrama de blocos do controlador de admitância com malha anti-windup.	42
Figura 3.11 - Diagrama de blocos do controlador P de velocidade.	43
Figura 3.12 - Diagrama de blocos do controlador PID de posição.	44
Figura 3.13 - Resposta do sistema real a uma solicitação em degrau em espaço livre.	45
Figura 3.14 - Resposta do sistema a um controlador de admitância de ganho unitário.	47
Figura 3.15 - Comportamento do sistema real para solicitações de velocidade reduzida.	47
Figura 3.16 - Resposta do sistema real em controlo por admitância com ganho 5.	48
Figura 3.17 - Instabilidade do sistema real com aumento de rigidez do ambiente.	49
Figura 3.18 - Referência de binário do motor para elevadas solicitações de velocidade.	50

Figura 4.1 – O <i>Steady Hand Robot</i>	54
Figura 4.2 - Os dois tipos de comportamento que o sistema que utiliza restrições virtuais pode apresentar: Ponto Alvo e Curva.....	56
Figura 4.3 - Tubo virtual definido por uma curva paramétrica.	59
Figura 4.4 - Cone virtual definido pelo seu ponto alvo e pela abertura.....	61
Figura 4.5 - Representação das diferentes zonas de admitância na restrição virtual do tipo Cone.	63
Figura 4.6 - Sequência de execução das rotinas da S-Function durante a simulação.	65
Figura 4.7 - Bloco <i>Simulink</i> que contém a restrição virtual Ponto Alvo.	66
Figura 4.8 - Simulação do comportamento do elemento terminal para uma restrição do tipo Ponto Alvo.....	68
Figura 4.9 - Forças tangencial e normal no movimento utilizando restrição do tipo Ponto Alvo.	69
Figura 4.10 - Referências de velocidade dadas aos controladores de eixo para uma restrição virtual do tipo Ponto Alvo.	69
Figura 4.11 – Simulação do comportamento do elemento terminal nos três eixos cartesianos para uma restrição do tipo Ponto Alvo.	70
Figura 4.12 - Bloco <i>Simulink</i> que contém a restrição virtual do tipo Curva.	72
Figura 4.13 - Exemplo de código em linguagem C para encontrar o ponto pertencente à restrição virtual mais próximo do elemento terminal.....	73
Figura 4.14 - Simulação do comportamento do elemento terminal para uma restrição virtual do tipo Curva.	74
Figura 4.15 - Erro de posição em relação à restrição virtual para uma restrição do tipo Curva..	75
Figura 4.16 - Forças tangencial e normal no movimento utilizando restrição do tipo Curva.....	76
Figura 4.17 – Erro de posição em relação à restrição virtual para uma restrição do tipo Curva para $c\tau = 0.1$	76
Figura 4.18 – Bloco <i>Simulink</i> que contém a restrição virtual do tipo Tubo.....	77
Figura 4.19 - Simulação do comportamento do elemento terminal para uma restrição virtual do tipo Tubo.	78
Figura 4.20 - Simulação do comportamento do elemento terminal para uma restrição do tipo Tubo, observada segundo a direção Z.	79
Figura 4.21 - Desvio de posição em relação a curva que define o centro da restrição virtual para uma restrição do tipo Tubo.....	79
Figura 4.22 - Bloco <i>Simulink</i> que contém a restrição do tipo Tubo Linear.	80
Figura 4.23 - Influência do parâmetro ϵ na restrição virtual Tubo.....	81
Figura 4.24 - Influência do parâmetro n na restrição virtual Tubo.	82
Figura 4.25 - Bloco do <i>Simulink</i> para a restrição virtual do tipo Cone.....	83
Figura 4.26 - Simulação do comportamento do elemento terminal para uma restrição virtual do tipo Cone.	83
Figura 5.1 - Posição do elemento terminal para uma restrição do tipo Ponto Alvo, movimentação no plano XY.	86
Figura 5.2 - Posição do elemento terminal para uma restrição do tipo Ponto Alvo, movimentação no espaço cartesiano.....	87
Figura 5.3 - Posição do elemento terminal para uma restrição do tipo Curva.....	88
Figura 5.4 - Erro de posição em relação à curva.	89

Figura 5.5 - Posição do elemento terminal para uma restrição virtual do tipo Tubo.	90
Figura 5.6 - Desvio de posição em relação ao centro da secção do tubo.	91
Figura 5.7 - Posição do elemento terminal para uma restrição virtual do tipo Tubo Linear.....	92
Figura 5.8 - Posição do elemento terminal para uma restrição virtual do tipo Cone em que o movimento se inicia dentro.	93
Figura 5.9 - Posição do elemento terminal para uma restrição virtual do tipo Cone em que o movimento se inicia fora.	94
Figura 5.10 - Teste de coordenação visual/motora.....	95
Figura 5.11 - Execução de uma restrição virtual do tipo Curva.	95
Figura 5.12 - Tentativa de seguimento de um contorno sem restrição virtual.	96

ÍNDICE DE TABELAS

Tabela 3.3.1 - Parâmetros utilizados na modelação de um sistema de transmissão com acoplamento de rigidez e amortecimento próprio.	33
Tabela 3.3.2 - Valores dos parâmetros que constituem o modelo.....	35
Tabela 3.3.3 - Ganhos dos controladores de posição PID.....	44

Capítulo 1

Introdução

1.1 Motivação

O trabalho apresentado nesta dissertação insere-se na área de desenvolvimento de sistemas de cooperação Homem-Máquina. Os primeiros sistemas robóticos tinham como objectivo substituir o homem em tarefas de montagem. Por exemplo, o *Programmable Universal Machine for Assembly* (PUMA) da *Unimation Inc.*, os sistemas da *Adept* e o IBM 7565 estão entre os primeiros sistemas robóticos articulados, implementados em grande escala, em aplicações de montagem. Estes sistemas operavam automaticamente e substituíam o homem em tarefas extenuantes, com vantagens ao nível da força disponível, repetibilidade e exactidão. No entanto estes sistemas não permitiam a coordenação de tarefas com o operador, quer por questões de segurança, quer tecnológicas. Só mais tarde surgem sistemas robóticos colaborativos, com os primeiros estudos a serem impulsionados por necessidades específicas na área militar e posteriormente na área médico-cirúrgica. Esta necessitava de dispositivos que auxiliassem os cirurgiões no manuseamento dos diversos instrumentos cirúrgicos permitindo reduzir o cansaço do cirurgião e aumentando a segurança da intervenção médica. A área médico-cirúrgica foi precursora da utilização de sistemas robóticos que implementam restrições virtuais. Nos últimos anos têm vindo a ser integrados nos diversos sistemas cirúrgicos, dispositivos que permitem o mapeamento 3D dos órgãos a operar permitindo assim reduzir os custos das intervenções cirúrgicas e os tempos de *setup* que estes dispositivos necessitavam. Actualmente vive-se um crescimento na implementação dos sistemas colaborativos, com estes a serem cada vez mais utilizados ao nível industrial, especialmente em operações que impliquem a deslocação e montagem de componentes pesados.

Neste capítulo é efectuada uma introdução aos sistemas colaborativos que incorporam restrições virtuais. Em primeiro lugar são apresentados estes dois conceitos: sistemas robóticos colaborativos e restrições virtuais. De seguida são apresentados dois paradigmas de controlo neste tipo de sistemas: admitância e impedância. Segue-se uma contextualização histórica do desenvolvimento deste tipo de sistemas e das suas principais características, sendo ainda referenciadas as suas principais vantagens e utilização nas diversas áreas.

Neste capítulo são também enumerados os objectivos deste projecto e feita uma descrição da estrutura da dissertação.

1.2 Sistemas Robóticos Colaborativos

Um sistema é considerado colaborativo quando existe uma interacção, em tempo-real, entre o operador e o robô durante a execução de uma tarefa (Peshkin M. *et al.* , 2001). A interacção entre o operador e o sistema visa reproduzir os movimentos físicos (membros superiores e/ou inferiores, mãos, dedos) do operador. Outros autores podem referir-se a estes sistemas como “sistemas cooperativos” (Marayong, 2007).

O termo colaborativo aplicado aos sistemas robóticos permite distinguir entre dois modelos: a telemanipulação e a manipulação colaborativa.

Na telemanipulação existe uma interacção indirecta entre o operador e o sistema robótico. O operador interage com um robô local (*master*), existindo um ou mais robôs remotos (*slaves*), responsáveis pela execução da tarefa. Como o operador não se encontra fisicamente no espaço de trabalho são necessários meios que proporcionem sensações hápticas e referências visuais.

Na manipulação colaborativa o operador encontra-se fisicamente no espaço de trabalho e interage directamente com o sistema robótico. Nesta interacção o robô reproduz em tempo-real os movimentos que o operador fisicamente impõe. Assim o operador mantém a

propriocepção¹ sobre o elemento terminal do robô, não sendo por isso necessários quaisquer outros equipamentos (Emeagwali, 2004).

É importante fazer-se a distinção entre estes dois sistemas e os “sistemas colaborativos de robôs”, que denotam uma cooperação entre dois ou mais robôs (Marayong, 2007).

Os sistemas colaborativos desenvolveram-se em complemento aos sistemas robóticos industriais os quais não podiam actuar em coordenação com os humanos por razões de segurança (Peshkin & Colgate, 1999).

Ao mover directamente o robô no espaço de trabalho o operador mantém a cinestesia (sentido pelo qual se tem a percepção dos membros e movimentos corporais), tornando os manipuladores colaborativos mais intuitivos de usar que os sistemas de telemanipulação. Porém, o facto de o operador actuar directamente a ferramenta aumenta a dificuldade no controlo. Os movimentos voluntários do operador são de difícil previsão e podem afectar significativamente a resposta do sistema. A adicionar a tudo isto, a mão humana também apresenta o seu próprio modelo cinemático o que pode resultar em indesejados movimentos involuntários (Wannasuphoprasit, 1999).

1.3 Restrições Virtuais

O conceito de restrição virtual (*virtual fixture* em inglês), corresponde à sobreposição de informação sensorial abstracta num dado espaço de trabalho com o objectivo de condicionar a resposta do sistema (Rosenberg, 1993). Este condicionamento aplicado aos sistemas colaborativos gera uma estimulação sensorial sobre o operador de modo a facilitar a execução de uma tarefa.

Este conceito é bastante lato e pressupõe que a informação sensorial não seja apenas posicional (posição/orientação no espaço) mas também visual e/ou auditiva. Porém, nesta dissertação, o termo “restrição virtual” significará a existência de superfícies ou curvas

¹ **Propriocepção** também denominada por cinestesia, é o termo utilizado para nomear a capacidade em reconhecer a localização espacial do corpo, a sua posição e orientação, a força exercida pelos músculos e a posição de cada parte do corpo em relação às demais, sem utilizar a visão. *em Wikipedia.org, 2010.*

modeladas (virtuais) que podem ter as propriedades das superfícies reais. Essas superfícies virtuais podem ter como objectivos impedir, ajudar ou dificultar a movimentação do robô em determinadas zonas do seu espaço de trabalho.

O conceito de restrição virtual pode ser descrito por uma metáfora, facilitando assim a sua compreensão. Considere-se a seguinte analogia com o mundo real, em que uma régua é usada para guiar o lápis de modo a traçar uma linha recta de forma rápida e precisa. Tal tarefa é de difícil execução para a maioria dos humanos sem a ajuda de guiamento. Assim o uso de uma régua permite guiar o lápis reduzindo o tremor da mão e o cansaço mental do operador, aumentando a qualidade de execução da linha recta.

Na Figura 1.1 é possível observar uma restrição virtual aplicada no seguimento de uma curva onde o elemento terminal apenas pode ser movimentado na direcção desejada.

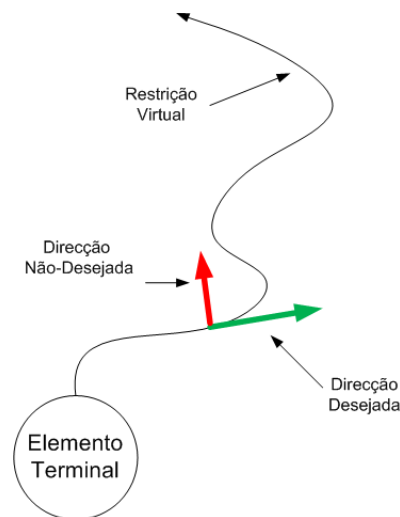


Figura 1.1 - Exemplo de uma restrição virtual aplicada ao seguimento de trajectórias.

Uma aplicação importante das restrições virtuais consiste na assistência em operações de montagem, especialmente quando os elementos são pesados e de grandes dimensões. Por exemplo, na indústria automóvel, mesmo existindo já dispositivos como compensadores pneumáticos e pontes rolantes, o operador ainda tem que exercer uma força considerável para poder manipular os componentes. Isso deve-se ao facto dos dispositivos não conseguirem lidar com a mudança de velocidade e direcção de componentes com elevada massa e também não conseguirem lidar com atritos existentes. Uma solução encontrada é usar uma superfície virtual, tal como um funil, que guie o componente para um dado ponto no espaço de trabalho. Este “funil virtual” tem dois propósitos: evitar colisões durante o processo, e guiar o operário

e a carga para o local de montagem (o operador pode assim fazer escorregar a peça sobre a superfície virtual) (Wannasuphoprasit, 1999).

As restrições virtuais são também aplicadas na área médica. Em cirurgias delicadas um robô pode ajudar a guiar a ferramenta do cirurgião para que este evite erros de posicionamento. Ainda na área médica, o uso de dispositivos que permitem acelerar o processo de reabilitação de pessoas que perderam movimentos nos membros inferiores, obrigando os pacientes a caminhar de forma correcta, recorre à utilização de restrições virtuais.

As restrições não se limitam a simular superfícies rígidas (sólidas), podem também simular superfícies com rigidez inferior (tipo mola linear/não linear), superfícies atractoras ou repulsivas e contactos de fricção.

Assim, os sistemas colaborativos que utilizam restrições virtuais podem ter as seguintes capacidades (Schneider, 2000):

- Localização: permitem dar ao operador a localização do elemento terminal em relação a um referencial (cinestesia), permitindo assim manter uma ligação entre o mundo real e o mundo virtual;
- Restrição de movimentos: permitem executar tarefas com restrições virtuais no espaço de trabalho (ex. seguimento de trajectórias, superfícies, etc.);
- Filtragem de movimentos: permitem a atenuação de efeitos de movimento não desejados (ex. tremuras);
- Controlo de força: permitem o controlo da força exercida pelo elemento terminal ou ferramenta (amplificação ou redução);
- Sincronização com acções externas: permitem a filtragem de movimentos externos à tarefa (ex. ondulação marítima, respiração);

A implementação das restrições virtuais em manipuladores colaborativos pode ser feita de diversas formas dependendo do tipo de actuadores e sistema de transmissão (Peshkin M. *et al.*, 2001). Assim nos manipuladores com transmissões reversíveis (*backdriven* em inglês) uma das abordagens é utilizar actuadores que se opõem à força do operador sempre que este tende a violar uma restrição virtual. Este tipo de robôs utiliza normalmente motores de accionamento directo ou com pequenas reduções. Nesta abordagem, o cumprimento da

restrição virtual está sempre limitado pela capacidade de força que os actuadores conseguem proporcionar, isto é, se a força do operador sobre o robô for maior que a força dos actuadores a restrição virtual não é garantida.

Nos manipuladores *non-backdriven* o operador controla o robô através da força exercida sobre um transdutor de força incorporado no elemento terminal, sendo que as restrições virtuais são concretizadas pela maior ou menor resposta dos actuadores. Estes robôs utilizam normalmente motores com grandes reduções. Deste modo o cumprimento da restrição virtual é garantido pela irreversibilidade das transmissões. Isto é, quando um manipulador contacta com uma superfície virtual os seus actuadores deixam de actuar. O manipulador só se deslocará para fora da restrição virtual se a força exercida pelo operador for suficiente para reverter o sentido das transmissões.

Existe ainda uma terceira abordagem que propõe o uso de travões, especialmente em aplicações que requerem uma rigidez continuamente variável na direcção normal à superfície virtual. Nesta abordagem os travões podem ser usados em combinação com motores (no caso de robôs irreversíveis) ou em substituição dos mesmos (robôs reversíveis). Neste último caso o robô torna-se inteiramente passivo, isto é, incapaz de gerar movimento próprio. A passividade tem vantagens óbvias na segurança e aceitação dos utilizadores. Nesta abordagem mais uma vez o cumprimento da restrição virtual é garantido pela capacidade de frenagem (neste caso contínua) dos travões. Assim, para um dado ponto no espaço de trabalho é dada uma referência de binário de frenagem. Se o operador suplantar essa referência o manipulador não cumprirá a restrição virtual.

Relativamente ao controlo implementado, na primeira abordagem o robô é controlado em impedância e na segunda em admitância. Na terceira abordagem recorre-se a um controlo híbrido no caso de o robô não poder ser guiado ou em impedância no caso de o robô ser completamente passivo. Os conceitos de controlo em admitância e em impedância serão abordados na secção 1.4 .

1.4 Controlo dos sistemas colaborativos (impedância vs admitância)

Existem duas grandes vertentes no controlo dos sistemas colaborativos com restrições virtuais: controlo por impedância e controlo por admitância.

No controlo por impedância o operador move o robô e este reage com uma força se necessário. Esta é a interacção básica entre o operador e o controlador. Vendo do lado da malha do controlador o paradigma é: entra posição – sai força, como está exemplificado na Figura 1.2 .

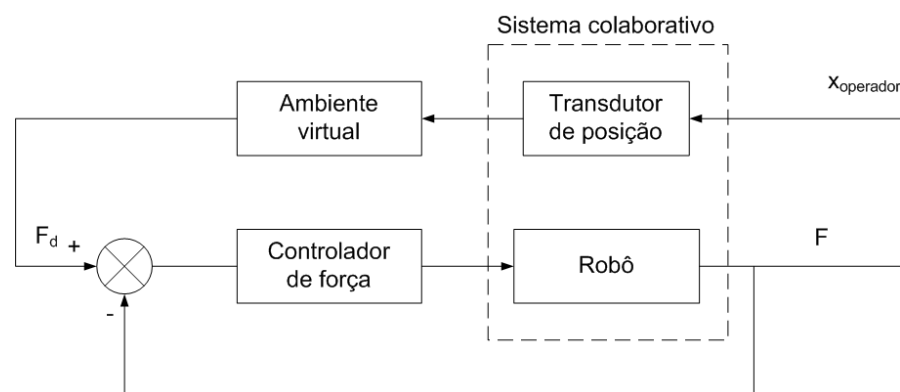


Figura 1. 2 - Controlo em impedância.

Num ambiente sem restrições virtuais, o utilizador é livre de mover o elemento terminal do manipulador robótico sem a influência dos motores. Em termos do controlo, o ganho entre a variação de posição e a força é zero. Isto mantém-se verdade mesmo que exista um contacto com uma superfície real. Não existe assim qualquer problema de estabilidade, uma vez que o operador pode parar o elemento terminal sem ser necessária qualquer reacção especial dos motores.

Porém, em contacto com superfícies virtuais existem problemas de estabilidade. Qualquer pequena alteração de posição na direcção normal à da superfície virtual vai provocar um aumento súbito na força de reacção dos motores. Isto implica um ganho muito grande entre a posição medida e o binário do motor. Por razões de estabilidade do sistema, esse ganho não pode ser infinitamente grande. Por isso, em controlo por impedância existe um limite máximo de rigidez que uma superfície virtual pode apresentar.

No controlo por admitância, o robô mede a força que o operador exerce sobre ele e reage com um movimento (aceleração, velocidade, posição). Vendo do lado da malha do controlador, o paradigma é: entra força – sai velocidade, posição ou aceleração, como está exemplificado na Figura 1.3.

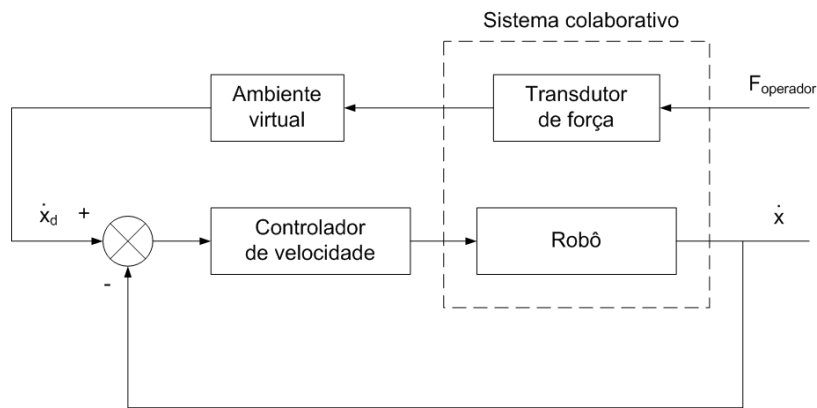


Figura 1.3 - Controlo em admitância.

Num ambiente livre, o robô deve acelerar rapidamente ao mais leve toque do operador. Isto significa que o ganho entre a força medida e a aceleração imposta deve ser elevado. Uma massa/rigidez virtual (simulada) pequena significa assim um ganho de controlo elevado. Podem assim existir problemas de estabilidade num ambiente livre quando a massa simulada é pequena. O mesmo pode acontecer no contacto com superfícies físicas rígidas. Um pequeno movimento do elemento terminal irá provocar um forte aumento nas forças de reacção. É o ambiente físico que fecha a malha de controlo, provocando um ganho elevado entre a posição e força levando à existência de instabilidade.

Porém, no contacto com superfícies virtuais, uma força elevada irá gerar um movimento de pequena amplitude. De facto no contacto com uma superfície infinitamente rígida o ganho entre a força e a posição é zero (Lammertse, 2004).

1.5 Desenvolvimento dos Sistemas Robóticos Colaborativos

A Arquimedes é atribuída a citação: “Dêem-me uma alavanca e um ponto de apoio e eu moverei o mundo”. Esta citação denota o grande problema que o Homem anterior à invenção

da alavanca tinha: mover sem esforço grandes massas. Foi esse mesmo problema que levou o Homem moderno a inventar novos mecanismos, mais leves, mais compactos e sobretudo mais seguros. Porém esses mecanismos (gruas, máquinas de arrasto, etc.) afastavam o homem da execução da tarefa e impediam que fossem executadas com exactidão. Neste contexto surge, nos anos 60, o conceito de amplificador humano (*man-amplifier* em inglês), representativo de um dispositivo que amplifica a força humana usando um sistema *master/slave*.

Em 1962 a Força Aérea Norte-Americana e o *Cornell Aeronautical Laboratory* iniciam uma investigação com vista ao desenvolvimento de equipamentos que permitissem aos soldados a capacidade de lidar com objectos pesados. Durante os estudos verificou-se que era impossível replicar todos os movimentos humanos. Em 1964 surge então o primeiro esboço de um braço, de accionamento hidráulico, com os graus de liberdade (GDL) necessários para executar grande parte das tarefas desejadas.

Em 1966 a *General Electric* apresenta o *Hardiman* (Figura 1.4), um sistema de dois exoesqueletos sobrepostos, usando a tipologia de controlo *master/slave*. O exoesqueleto interno, *master*, era controlado pelos movimentos do operador e o exoesqueleto externo, *slave*, de actuação hidráulica, respondia à actuação do primeiro.



Figura 1.4 - *Hardiman* construído pela *General Electric* (Szondy).

Em 1993, Kazerooni introduz o conceito de extensões (*extenders* em inglês) e distingue os dispositivos *master/slave* dos *extenders*. Nestes o operador controla directamente em força

o dispositivo (não existe arquitectura *master/slave*, *joysticks*, exoesqueletos...) (Kazerooni, 1993). Ainda em 1993, Rosenberg introduz o conceito de restrição virtual (*virtual fixture* em inglês) (Rosenberg, 1993).

Em Troccaz & Delnondedieu (1996) é descrito o funcionamento do PADyC (*Passive Arm with Dynamic Constrains* em inglês) representado na Figura 1.5 . Trata-se de um equipamento com características que potenciam a sua aplicação na área cirúrgica. É um sistema com 2 GDL em que a velocidade de movimentação pode ser definida entre dois intervalos de velocidades de referência. Quando não está em contacto com uma superfície virtual o sistema permite que sejam utilizadas quaisquer velocidades num intervalo definido. Quando em contacto com uma superfície virtual, as velocidades de referência tendem para zero na direcção perpendicular à superfície. Uma limitação deste dispositivo é não permitir ter a sensação de fricção e suavidade de movimento.

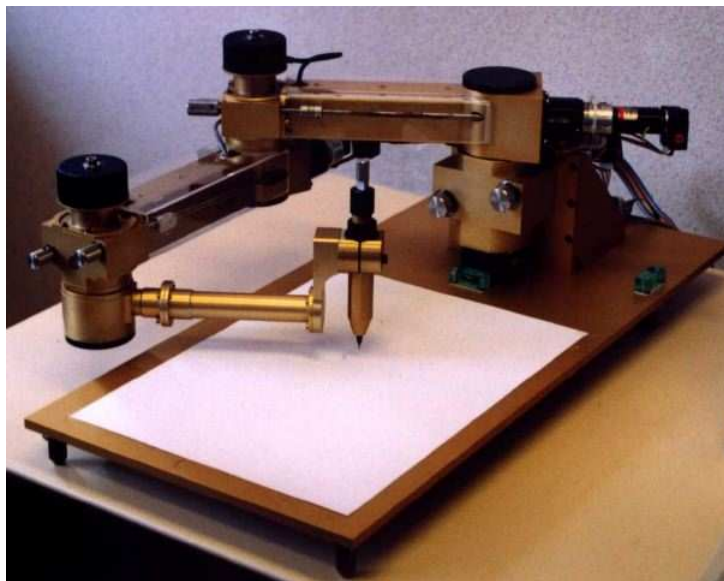


Figura 1.5 - Dispositivo PADyC (Laboratory for Intelligent Mechanical Systems, 2004).

Em Book *et al.* (1996) é descrito o funcionamento do P-TER (*Passive Trajectory Enhancing Robot* em inglês) representado na Figura 1.6. Trata-se de um equipamento com 2 GDL e com um espaço de trabalho planar horizontal. O P-TER utiliza transmissões, que permitem variar o comprimento dos seus elos, e travões que permitem abrandar os movimentos. O sistema sofreu melhoramentos posteriores no sistema de travões mas continua a apresentar um funcionamento irregular.

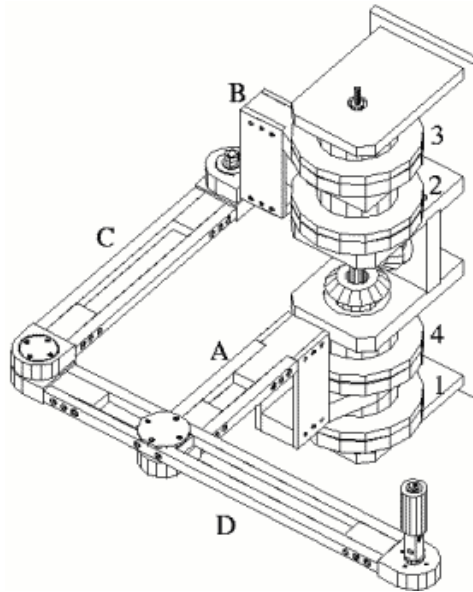


Figura 1.6 - Dispositivo P-TER (Intelligent Machine Dynamics Laboratory).

Peshkin e Colgate (Peshkin & Colgate, 1999), (Peshkin *et al.*, 2001) têm desenvolvido uma série de equipamentos utilizando transmissões de variação contínua (CVT) a que dão o nome de COBOT (*Collaborative Robot* em inglês).

O primeiro equipamento representado na Figura 1.7, é constituído por um monociclo, apoiado em guias, que permite movimentos no plano horizontal. Apesar da sua simplicidade, este equipamento demonstra dois tipos de funcionamento: um livre que permite o movimento em qualquer direcção, e um usando restrições virtuais, em que o equipamento pode ser guiado ao longo de uma trajectória definida por software. No caso de o operador movimentar o elemento terminal em direcção a uma superfície virtual este opõe-se ao movimento, rodando a roda na direcção tangente à superfície.



Figura 1.7 – Dispositivo COBOT monociclo (Intelligent Machine Dynamics Laboratory).

Um segundo equipamento representado na Figura 1.8 é um triciclo, apelidado de “scooter”. De funcionamento semelhante ao primeiro equipamento, este prescinde do sistema de guias, tendo assim uma área de trabalho mais ampla.

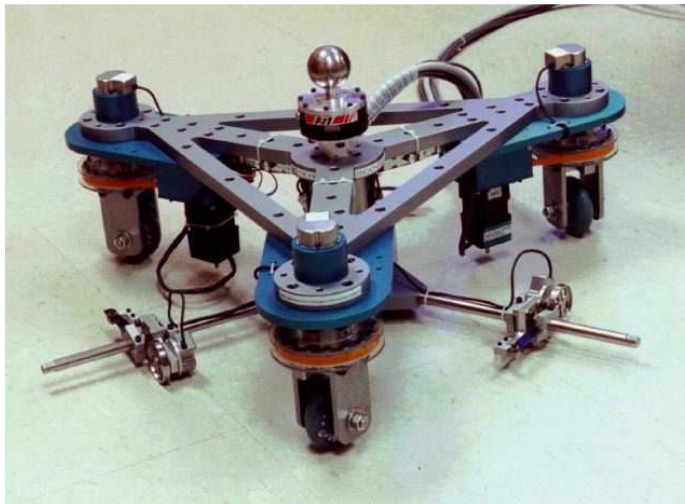


Figura 1.8 - Dispositivo COBOT triciclo "scooter" (Laboratory for Intelligent Mechanical Systems, 2004).

Em 1995, a *General Motors Foundation* financia o desenvolvimento de robôs de forma a minimizar os efeitos de algumas tarefas mais exigentes sobre os seus trabalhadores, os *Intelligent Assist Devices* (IAD's) (Colgate *et al.*, 2003). Surge assim um primeiro dispositivo que permite manobrar as portas dos automóveis guiando-as até ao local de montagem (Figura

1.9). Destes estudos surgem ainda outros dispositivos também de apoio à produção automóvel.

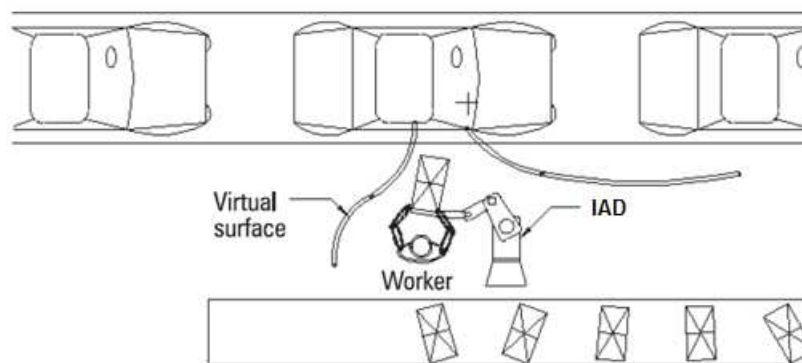


Figura 1.9 - Exemplo da utilização de IAD's na montagem de componentes na indústria automóvel (Peshkin & Colgate, 1999).

Durante a década de 90 surgem duas novas áreas, a *Computer Assisted Surgery* (CAS) e a *Computer Assisted Therapy* (CAT) impulsionadas pela necessidade de aumentar a segurança em intervenções médicas delicadas, minimizar as tarefas invasivas e melhorar a reaprendizagem de movimentos durante o período de reabilitação. No âmbito destas áreas surgiram variadíssimos dispositivos médicos baseados nos sistemas até aí desenvolvidos.

O *Lokomat*® (Figura 1.10), produzido pela *Hocoma*, é um exemplo da utilização dos sistemas colaborativos usando restrições virtuais na área médica. Este equipamento é constituído por um exoesqueleto motorizado que permite simular o caminhar a doentes com diferentes graus de deficiência motora.



Figura 1.10 - Dispositivo Lokomat® (Hocoma).

Um outro sistema com aplicação na área médica é o *ACROBOT*® (Figura 1.11). Este equipamento desenvolvido pelo *Imperial College* está vocacionado para ajudar o cirurgião em operações ao joelho, impedindo-o de fazer um desbaste exagerado no fémur.



Figura 1.11 - Uso do ACROBOT em cirurgia ortopédica (Davies *et al.*, 2004).

1.6 Objectivos da dissertação

Este trabalho tem por objectivo o desenvolvimento de uma estratégia de controlo para a implementação de restrições virtuais num robô cartesiano de três eixos.

Pretende-se assim que um operador possa controlar directamente e sem esforço, o elemento terminal, enquanto que o sistema robótico pode providenciar amplificação/atenuação de força e restrições virtuais.

O operador deve poder especificar quais as zonas o espaço de trabalho onde terá total liberdade de movimentos e as zonas onde o movimento do elemento terminal será restringido (guiado, dificultado ou proibido).

O trabalho apresentará diversas tarefas, como a integração do transdutor de força no robô cartesiano, o desenvolvimento e implementação da estratégia de controlo, a definição de um programa experimental e a avaliação do desempenho do dispositivo.

Do trabalho deverá assim resultar um protótipo, que pode ser usado em demonstrações, bem como permitir a continuação da investigação nesta área do controlo por admitância/impedância. O trabalho deverá assim permitir a demonstração de uma possível aplicação prática deste tipo de tecnologias em outras áreas (cirúrgica, industrial, transportes...)

1.7 Estrutura da dissertação

O presente documento é composto por 6 capítulos. Tendo em conta os objectivos enunciados na secção anterior, pretende-se detalhar com precisão o trabalho realizado, permitindo e facilitando a utilização futura, por terceiros, dos algoritmos aqui desenvolvidos e implementados.

Assim, no Capítulo 1 dá-se a introdução ao projecto, expondo a motivação deste trabalho. São introduzidos os conceitos de manipulação colaborativa e de restrição virtual, explicadas as duas grandes vertentes de controlo na área em causa e é apresentada uma súmula do desenvolvimento dos sistemas robóticos colaborativos, uma breve revisão de pelos trabalhos que serviram de ponto de partida para este projecto.

No Capítulo 2 são apresentadas as principais características do robô cartesiano. É feita uma breve descrição do software utilizado para construir a aplicação de controlo e é feita a descrição da arquitectura de controlo utilizada para o comando do sistema robótico. Neste capítulo é também descrita a concepção da interface física do operador com o elemento terminal do robô.

No Capítulo 3 inicia com a descrição comportamental do sistema. É descrito o modelo do eixo linear utilizado e são apresentados os diferentes controladores utilizados pelo sistema, com especial ênfase para o controlador de velocidade I-P. Por último é feita uma análise experimental do controlo por admitância.

No Capítulo 4 é feita uma descrição do modelo de restrições virtuais aplicado, são descritos os casos em estudo e é descrita a sua implementação na aplicação de controlo. As restrições são simuladas recorrendo ao modelo do eixo linear descrito no Capítulo 3.

No Capítulo 5 é feita uma análise ao comportamento das restrições virtuais implementadas. Dada a natureza do sistema a análise não é apenas quantitativa mas sim qualitativa. Tentam-se identificar as causas para os comportamentos díspares à simulação.

Finalmente no Capítulo 6 apresentam-se as principais conclusões e possíveis desenvolvimentos futuros.

1.8 Conclusões

Neste capítulo foi apresentada uma introdução aos sistemas robóticos colaborativos e às restrições virtuais, descrevendo-se a sua origem, significado, áreas de aplicação e tecnologias utilizadas. Foram ainda apresentados alguns exemplos de trabalhos desenvolvidos na área dos sistemas colaborativos.

Do estudo efectuado dos sistemas colaborativos robóticos que utilizam restrições virtuais verificou-se que existem diversas áreas de aplicação. Essas áreas tiram proveito das vantagens destes sistemas para a realização de tarefas que envolvem cargas pesadas e volumosas ou cargas pequenas mas movimentações exactas, que são de difícil execução pelo homem.

Neste capítulo foram ainda introduzidas as diferentes estratégias de controlo que são normalmente implementadas neste tipo de sistemas. Por último foram enunciados os objectivos da dissertação e feita uma descrição da sua estrutura.

Capítulo 2

Descrição dos meios utilizados

2.1. Introdução

Neste capítulo é efectuada a descrição das características principais do robô cartesiano e dos seus diversos componentes (actuação, sensorização e segurança). É descrita a aplicação de simulação e controlo do sistema robótico (*Matlab/Simulink*) e é feita uma abordagem à arquitectura de controlo usada no equipamento.

Este capítulo aborda também o trabalho efectuado na concepção de novas peças necessárias para a utilização do robô cartesiano. Estas peças têm como função a interface entre o operador e o elemento terminal do robô e por isso foram concebidas com propósitos de ergonomia e segurança.

2.2. Robô cartesiano

O robô cartesiano existente no Laboratório de Robótica do Departamento de Engenharia Mecânica é composto por três eixos lineares de fusos de esferas da *Rexroth/Bosh* utilizando uma configuração física em pórtico, conforme pode ser visto na Figura 2.1.

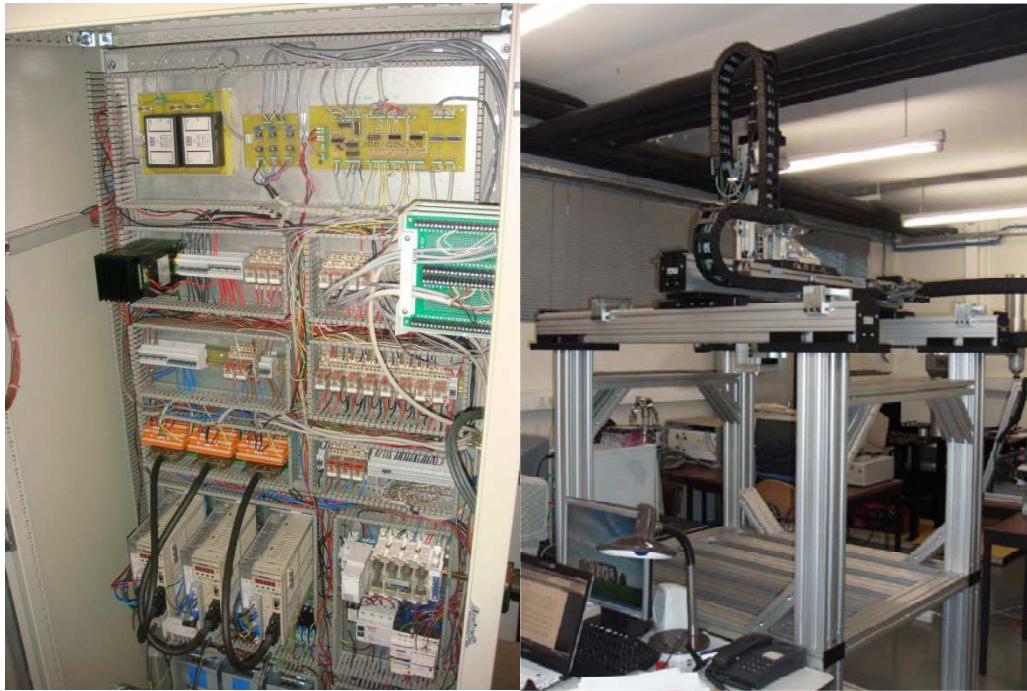


Figura 2.1 - Robô cartesiano utilizado neste trabalho.

O movimento do eixo X é realizado por dois eixos paralelos, um de guiamento e um motor. Sobre os patins destes dois eixos está montado o eixo Y, cujo patim suporta o eixo Z.

O robô apresenta um atravancamento de 1750x1550x2900 mm, e um volume de trabalho de 1080x1250x720 mm. Neste projecto, devido à presença de elementos adicionais e por questões de segurança, o volume de trabalho é de 630x520x205mm.

Existem fins-de-curso mecânicos de posição montados nas extremidades de cada eixo, que permitem limitar o espaço de trabalho do manipulador. Assim, quando estes são actuados enviam um sinal ao *driver* de potência dos motores para que estes parem. Para além dos fins-de-curso mecânicos, existem batentes mecânicos de protecção em cada extremidade de cada um dos eixos.

Existem ainda detectores indutivos que estão localizados aproximadamente a meio do curso de cada eixo e que são utilizados para definir o zero de cada eixo.

Os servomotores que movem os eixos lineares são do tipo *brushless* AC da *Omron* (R88M-W75030) de 750 W de potência. Os motores são idênticos em todos os três eixos

tendo o motor do eixo Z acoplado um travão electromagnético. Sendo os motores de 3000 rpm, as velocidades nominais dos eixos são de 1 m/s em X e Y e 0.5 m/s em Z, atendendo ao passo dos fusos utilizados (eixos X e Y, 20mm; eixo Z, 10mm).

Aos motores estão associados *drivers* da *Omron* (R88D-WT08HH) e *encoders* (*Omron*) incrementais standard para feedback de posição, que se ligam aos *drivers* através de uma comunicação série proprietária.

O transdutor de força de 6 eixos, disponível para integração no sistema, é da *JR3*, (modelo 67M25A-I40). Está montado entre o elemento terminal do robô e o suporte do punho do operador. Este transdutor pode ler cargas até 200 N em X e Y, 400 N em Z e momentos até 12.5 N.m.

A aplicação de controlo criada para este robô cartesiano corre em ambiente *Matlab/Simulink/xPC Target*.

2.3. O Matlab/Simulink

O *Matlab* (acrónimo de "MATrix LABoratory") é uma aplicação informática que disponibiliza um ambiente de computação numérica criado por Cleve Moler na década de 70. Desenvolvido pela *MathWorks*, o *Matlab* é especialmente adequado para a manipulação de matrizes, processamento de sinais e construção de gráficos. A sua grande vantagem é poder expressar os problemas da forma como eles são escritos matematicamente e não como na programação tradicional onde, por exemplo, a multiplicação de vectores é programada operação a operação.

O sistema foi inicialmente usado na área de projecto de controlo, mas rapidamente se espalhou a outras áreas muito graças às diversas *toolbox* criadas. É actualmente uma das ferramentas preferenciais no meio científico e no ensino de disciplinas de álgebra, análise numérica e elementos finitos.

O *Matlab* pode usar funções e subrotinas escritas em linguagem de programação C ou *Fortran*. É necessário para isso criar funções invólucro (*wrapper functions*) que permitem trocas de informação com o *Matlab*. Esses ficheiros são depois compilados, criando assim ficheiros MEX. Os ficheiros MEX (*Matlab EXcutable*) são objectos que podem ser utilizados dinamicamente pelo *Matlab*, permitindo a este invocar código exterior como se tratasse de uma função nativa.

O *Simulink* é uma ferramenta desenvolvida pela *MathWorks* para modelação, simulação e análise de sistemas dinâmicos. É essencialmente usado nas áreas de teoria de controlo e processamento de sinal embora encontre outras aplicações como na modelação de processos produtivos e validação de resultados. Trata-se de uma ferramenta de programação em ambiente gráfico composta por um conjunto de bibliotecas de funções que podem ser configuradas pelo programador. O programador pode criar as suas próprias bibliotecas ou adicionar bibliotecas de terceiros.

Uma das aplicações do *Simulink*, importante no desenvolvimento da estratégia de controlo, é o *Stateflow* que disponibiliza ao programador uma interface gráfica capaz de desenvolver e simular sistemas comandados por eventos. O programador pode criar um diagrama com vários estados e associar a estes comportamentos diferentes do sistema controlado. Esta aplicação permite ainda definir as transições entre os diversos estados por associação de eventos ocorridos fora ou dentro da aplicação de controlo.

Outra das aplicações do *Simulink* é o *Real-Time Workshop*, que permite gerar de forma automática código C (código fonte) para aplicações em tempo-real. Esta aplicação contém compiladores para diferentes arquitecturas de hardware e software. Uma dessas arquitecturas é a *xPC Target*, uma solução *stand-alone*, que permite ao programador o desenvolvimento de toda a estratégia num *PC* (*Personal Computer*), ao qual se dá o nome de *Host PC*, e posteriormente descarregar a aplicação de controlo para o *Target PC*. O *Target PC* tem como função principal a execução em tempo-real da aplicação de controlo. Existem outras aplicações entre as quais a *Real-Time Windows Target*, que é um módulo em tempo real que pode correr sobre um sistema operativo *Windows*. Uma grande vantagem do *Real Time Workshop* é o facto de este disponibilizar uma grande variedade de *device drivers* de tempo-real que permitem uma interface com grande parte das cartas I/O, A/D e D/A disponíveis no

mercado. Para além disso o programador tem sempre a possibilidade de criar os seus próprios *device drivers*.

2.4. A arquitectura Host/Target PC

O robô cartesiano apresenta uma arquitectura de controlo com uma configuração *Host/Target*, baseada em computadores pessoais. O *Host PC* funciona como plataforma de desenvolvimento, bem como centro de supervisão e comando. O *Target PC* desempenha o papel de controlador do sistema robótico. O *Host PC* comunica com o *Target PC* através de uma ligação *ethernet* dedicada (cabo *crossover*) (Cardoso, 2009). A implementação da arquitectura *Host/Target PC* está exemplificada na Figura 2.2 :

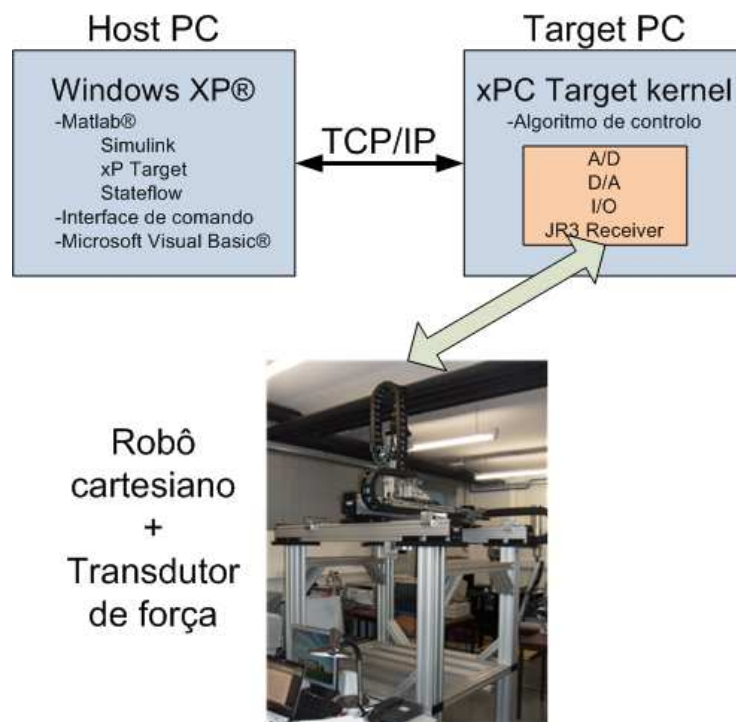


Figura 2.2 - Arquitectura *HOST/TARGET PC*.

As funções de controlador do robô cartesiano são asseguradas por um *Desktop PC*, *Pentium IV @ 1.71 GHz* (*Target PC*). Neste *PC* estão instaladas cartas de aquisição de dados, que possuem uma interface para barramento *PCI* (*Peripheral Component Interconnect*). Estas cartas de aquisição são responsáveis pela interface do computador com o sistema físico a

controlar. É pois, através delas que é possível o comando do robô e a monitorização e controlo das variáveis analógicas e sinais digitais.

O *Target PC* corre um kernel de tempo-real *xPC Target* do *Matlab/Simulink*. Assim, os recursos do *Target PC* são unicamente utilizados para a realização de tarefas de controlo. Sendo essas tarefas executadas com um tempo de ciclo de 1 ms (1 kHz).

O *Host PC* é um *Desktop PC Pentium IV @ 2.8 GHz* e corre o sistema operativo *Windows XP Professional*, o software de desenvolvimento do controlador do sistema *Matlab/Simulink* e o software de desenvolvimento de ambientes gráficos *Microsoft Visual Studio*.

2.5. Desenho e concepção do sistema de interface robô/operador

Para ser possível a integração do transdutor de força/momento no robô foi necessária a concepção de novas peças. Assim as peças necessárias para a realização deste trabalho são:

- Interface robô - transdutor;
- Interface transdutor - porta-ferramentas;
- Adaptador do punho;
- Punho.

A interface robô – transdutor (Figura 2.3) visa substituir a actual interface do robô que se concluiu ser inadequada à integração do transdutor. Durante o desenvolvimento das novas peças surgiram duas abordagens: uma propunha a concepção de um adaptador entre a peça existente e o transdutor, a outra propunha a concepção de uma interface completamente nova. Dadas as características de ligação com o transdutor optou-se pela segunda abordagem, visto que a primeira obrigava à concepção de duas novas peças o que tornaria o custo de maquinaria muito superior. O material utilizado na fabricação desta peça é alumínio.

Os benefícios desta nova interface são grandes, visto que se trata de uma interface normalizada ISO 9409-1 – I40, permitindo assim a integração de outros componentes compatíveis.

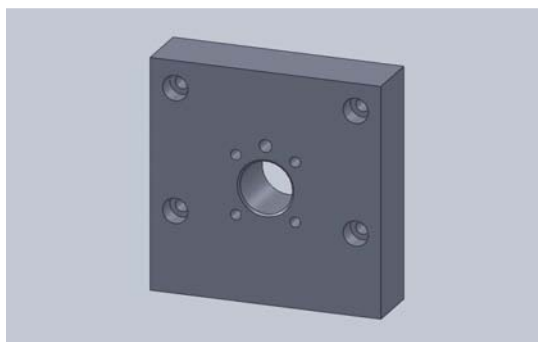


Figura 2.3 - Nova interface do robô.

A interface transdutor – porta-ferramentas (Figura 2.4) permite o suporte do punho para além de servir de elemento de ligação entre o sistema de porta ferramentas (GWS-64 da *Schunk*) e o transdutor. Esta peça tem como objectivos: ser leve, evitando assim uma carga fixa sobre o transdutor; ser compacta de modo a não reduzir muito o espaço de trabalho disponibilizado; permitir a integração de um punho em qualquer uma das suas faces laterais disponíveis de modo a que operador possa montar o punho na posição que mais lhe convier. O material utilizado na fabricação desta peça é alumínio.

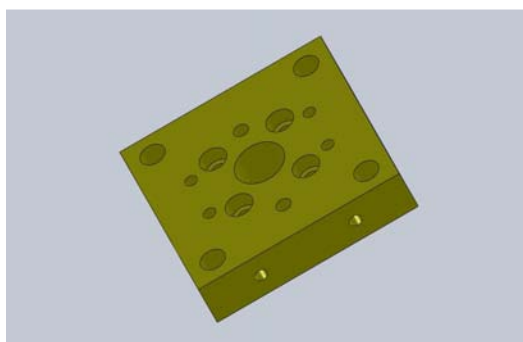


Figura 2.4 - Interface entre o transdutor de força e o sistema porta-ferramentas.

O adaptador do punho (Figura 2.5) é uma peça que visa a ligação entre a interface transdutor – porta-ferramentas e o punho. Esta peça necessita ser compacta e ser capaz de integrar a maioria dos punhos comerciais de ligação roscada M8, assim como ser também capaz de integrar o punho disponível. O material utilizado na fabricação desta peça é alumínio.

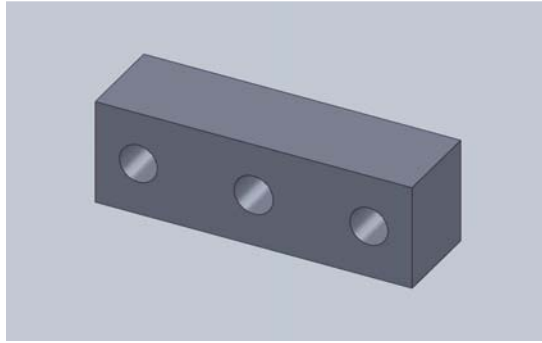


Figura 2.5 - Adaptador para os diversos punhos.

O punho pretende ser a peça de interface com o operador e é constituída por dois elementos: um núcleo (Figura 2.6) e uma carcaça (Figura 2.7). O núcleo serve de suporte à carcaça e possui uma interface roscada, M8, para permitir a ligação ao adaptador do punho. Esta peça será fabricada a partir de um varão de aço com 16 mm de diâmetro. A carcaça tem como requisito ser ergonómica e confortável para o operador e possuir botões que permitam a actuação de uma ferramenta e/ou o bloqueamento do robô. Esta carcaça foi recuperada de um *joystick* fabricado pela *QuickShot*.



Figura 2.6 - Suporte para o punho.



Figura 2.7 - Carcaça ergonómica para integração no punho.

A interface do elemento terminal com o utilizador pode ser visualizada na Figura 2.8 :

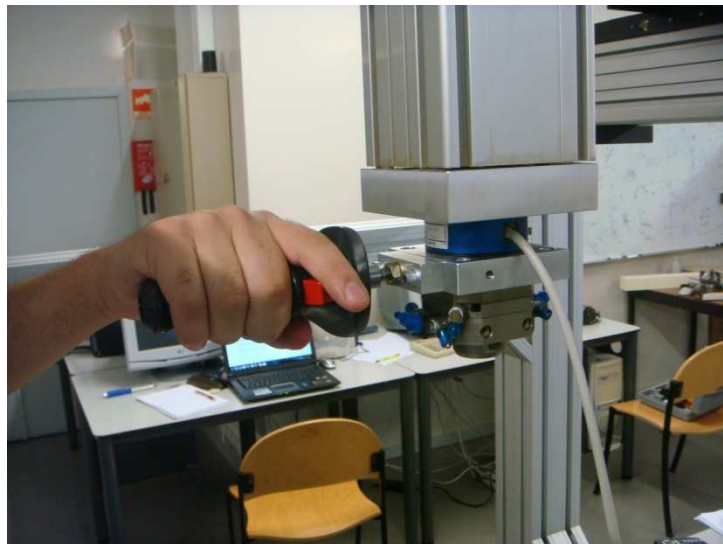


Figura 2.8 - Uso do punho.

2.6. Conclusões

Neste capítulo foi efectuada uma descrição do robô cartesiano, foram caracterizados os seus componentes essenciais. Foi ainda descrito o espaço de trabalho disponível para operador.

Foi introduzida a aplicação de simulação e controlo *Matlab/Simulink*. Posteriormente foi abordada a arquitectura de comando *Host/Target* e foram caracterizados os meios físicos e de software que a suportam.

Por último foi também apresentado o processo de concepção das novas peças necessárias à execução deste trabalho, evidenciando os seus requisitos técnicos e de operação.

Capítulo 3

Descrição da estratégia de controlo do robô cartesiano

3.1 Introdução

O software de controlo do sistema robótico foi desenvolvido em *Matlab/Simulink*. Esta poderosa ferramenta permite modelar, simular e implementar, não só o controlo dinâmico dos eixos, como o controlo funcional de todo o sistema.

Assim, este capítulo inicia com a descrição comportamental do sistema robótico colaborativo. Esta descrição sumária permite conhecer a resposta do sistema às diferentes receptividades internas e externas.

Seguidamente é descrito o modelo do eixo linear utilizado para simulação, tanto completo como simplificado.

São também apresentados os diferentes controladores utilizados pelo sistema. A existência de diferentes controladores deve-se ao facto de se pretender dispor de diferentes respostas em função das operações a realizar com o manipulador. Assim, se para um controlo autónomo, em que se pretende exactidão no posicionamento, é utilizado um controlador de posição PID (proporcional – integral – derivativo), já no controlo por admitância, em que o importante é uma boa interacção Homem-Máquina, é utilizado um controlador de velocidade I-P (integral - proporcional). Para a referenciação do elemento terminal é utilizado um controlador P (proporcional). É também descrito neste capítulo o ajuste dos parâmetros dos controladores assim como a utilização das malhas *anti-windup*.

Por último é feita uma análise experimental do controlador de admitância implementado, tendo por base as simulações obtidas utilizando os modelos anteriormente descritos. Os controladores de posição PID e de velocidade P descritos neste capítulo já foram objecto de estudo num trabalho anterior usando este mesmo sistema robótico (Cardoso, 2009), pelo que foi omitida a análise experimental dos mesmos.

3.2 Máquina de estados - *Stateflow*

O *Stateflow* é uma ferramenta, desenvolvida para o *Matlab/Simulink*, que permite a criação e simulação de sistemas de eventos discretos. Esta providencia os elementos necessários para descrever de uma forma simples e compreensível o funcionamento de um qualquer sistema complexo.

Na presente abordagem ao controlo do robô foi incluído um único diagrama de estados que contempla todos os modos de funcionamento do robô (Figura 3.1), associando a cada estado um bloco *Simulink*. As transições do diagrama de estados estão associadas a receptividades que podem ser eventos ou condições. Às transições podem ainda estar associadas acções transitórias que são executadas sempre que se dá tal transição. Existem dois tipos de receptividades: as locais, que podem ser geradas por *software* e as externas à máquina de estados, correspondentes a entradas geradas por hardware, ou geradas pela interface gráfica ou pelo término de determinadas acções (*flags*). As entradas da *Máquina de Estados* não se limitam apenas a sinais lógicos (*On/Off*). Assim é possível armazenar dados tais como o tempo decorrente da execução e as posições lineares do sistema robótico.

As saídas da *Máquina de Estados* permitem não só activar/desactivar blocos do *Simulink* como também disponibilizar dados relativos a tempos de execução e posições lineares.

Na Figura 3.1 está representado o diagrama de estados implementado no sistema robótico colaborativo utilizando restrições virtuais.

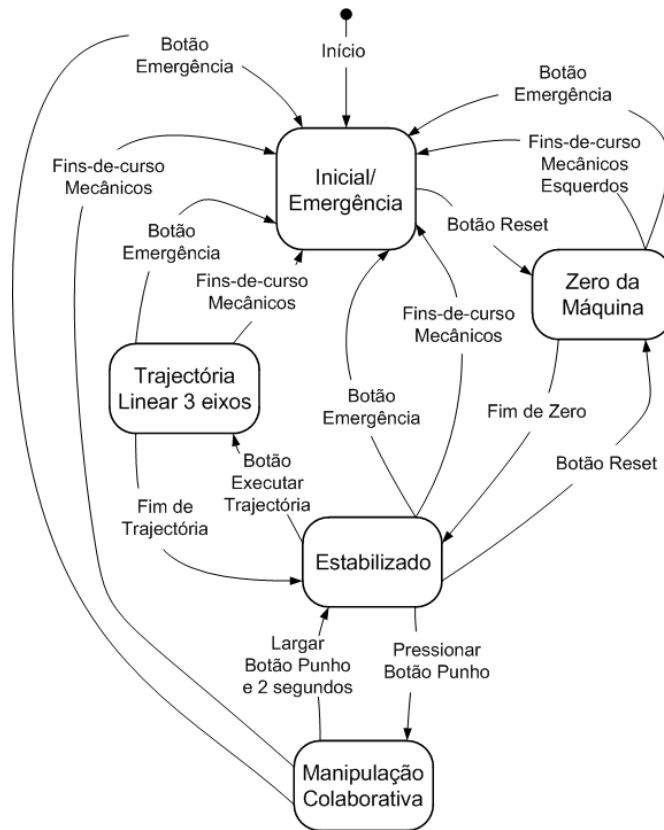


Figura 3.1 - Diagrama da Máquina de Estados do sistema robótico.

Quando o robô é ligado (activação do *software* de controlo) o estado “Inicial/Emergência” fica imediatamente activo. Neste estado activa-se um bloco *Simulink* onde são activados os sinais digitais *Run* e *Brake*. Deste modo é desactivado o travão eléctrico do eixo Z ficando assim este eixo livre de se mover e são activados os *drivers* de potência, sendo dadas referências nulas aos motores à excepção do eixo Z onde é dada uma referência que compensa o movimento gerado pelo peso próprio do referido eixo.

Só é possível sair do estado inicial quando for dada a ordem para se efectuar o procedimento designado “Zero da Máquina”. Este procedimento faz o zero dos *encoders*, terminando com elemento terminal do manipulador no ponto (0,0,0). Todos os movimentos efectuados após este procedimento serão relativos a esse ponto. Desta forma este procedimento é obrigatório numa fase de inicialização, não sendo possível transitar para qualquer outro estado que permita o controlo de movimentos enquanto tal procedimento não for finalizado. O procedimento inicia-se se e só se o sistema se encontrar no estado “Inicial/Emergência” e o operador accionar o botão *Reset* na interface de comando.

O procedimento “Zero da Máquina” está associado a um *super-estado* que é composto por quatro sub-estados e que se encontra caracterizado na Figura 3.2, executados em paralelo a partir do momento em que o procedimento é iniciado. Os três primeiros sub-estados estão associados a subsistemas *Simulink* que fornecem referências de velocidade aos controladores de velocidade P de cada eixo. Estas referências alteram-se mediante o estado que se encontra activo. Estado esse determinado pelas receptividades provenientes dos detectores de fins-de-curso e indutivos.

O procedimento é executado obedecendo a uma sequência de operações. Na primeira sequência são dadas referências de velocidade de modo a que os eixos se movimentem num determinado sentido. Quando cada eixo chega ao final do seu curso, a referência dada troca de sinal e o eixo movimenta-se no sentido contrário. Sendo activados os fins de curso mecânicos, o sinal *Overtravel* ficará activo até ao final de todo o procedimento. Após ser invertido o movimento do eixo, no momento em que ocorre uma mudança de estado (flanco descendente) do detector indutivo é atribuída uma referência nula ao controlador de velocidade P, ou seja, fica imobilizado nessa posição, sendo feita a inicialização dos *encoders*.

Esta sequência garante que, independentemente da posição inicial do elemento terminal, no decurso do procedimento todos os eixos activarão os detectores indutivos pelo menos uma vez. Desse modo o *hardware* do robô pode sempre efectuar o *zero* dos *encoders* incrementais.

No final deste procedimento ocorre ainda o *reset* ao sinal de *Overtravel*. Só depois disso é que ocorre a transição do controlador de velocidade P para o controlador posição PID.

Se durante o procedimento for activado um dos fins-de-curso mecânicos colocados no lado oposto ao movimento pretendido, o sistema retornará ao estado *Inicial/Emergência* como medida de segurança.

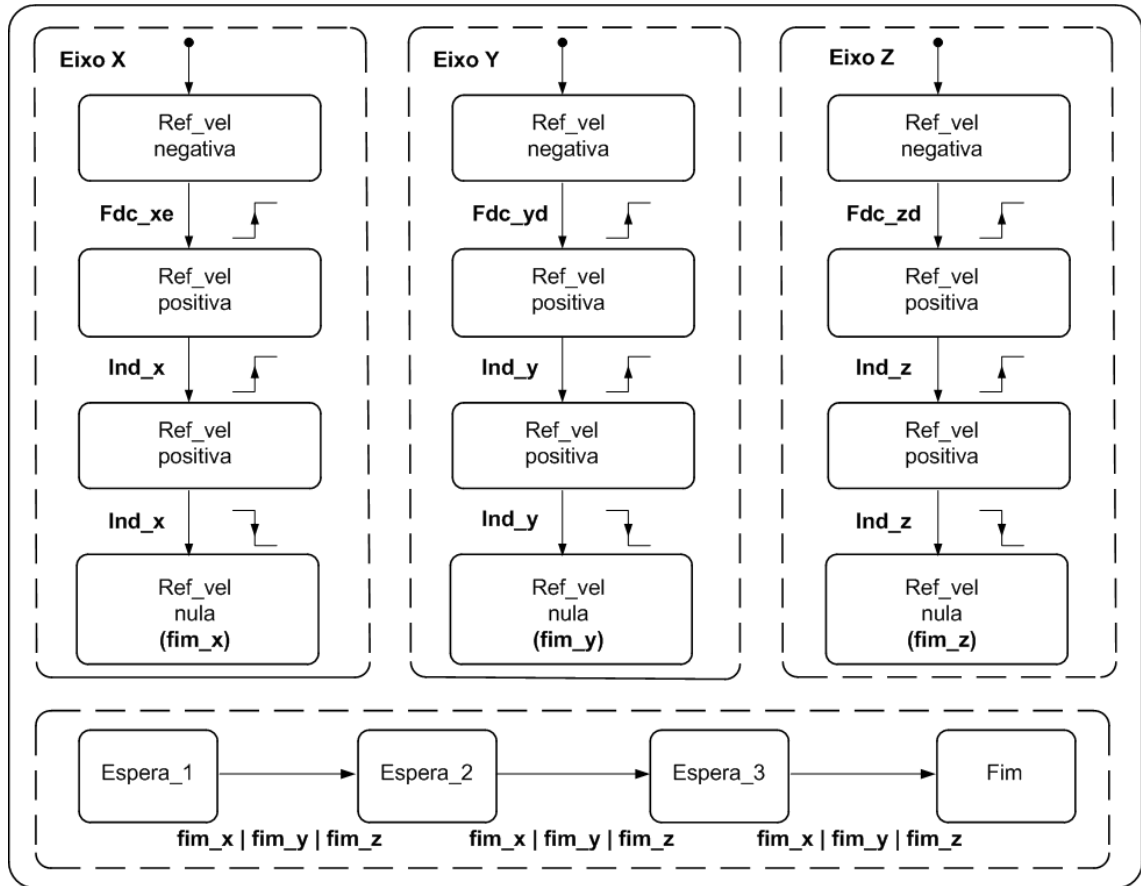


Figura 3.2 - Diagrama do super-estado: Zero da Máquina.

Findo o procedimento *Zero da Máquina*, a máquina de estados transita para o estado *Estabilizado*. Este estado faz activar um subsistema *Simulink* que contém controladores de posição PID, atribuindo referências fixas aos mesmos. Essas referências correspondem às coordenadas do elemento terminal no momento em que ocorre a transição para o estado *Estabilizado*. Isto é, o sistema fica parado em controlo de posição.

Quando está no estado *Estabilizado*, o sistema só evolui para outro estado por acção do operador. Este pode interagir com a interface gráfica (botão *Executar Trajectória*), ou por acção sobre os elementos físicos (botoneira de emergência ou botão do punho).

Do estado *Estabilizado* o sistema pode evoluir para o estado *Trajectória Linear 3 eixos* onde é possível executar uma trajectória linear no espaço 3D, uma vez que é efectuada com os três eixos em movimento. A posição final do elemento terminal é definida pelo operador através da interface gráfica antes de se iniciar o procedimento. A este estado estão associados vários subsistemas, entre os quais um gerador de trajectórias 3D e os controladores de posição

PID. O funcionamento dos subsistemas *Simulink* associados a este estado estão descritos em (Cardoso, 2009).

Caso o operador pretenda controlar directamente o elemento terminal deverá pressionar o botão de comando do punho. Desta forma o sistema transita do controlo de posição para o controlo de admitância/velocidade I-P por intermédio da activação de um subsistema *Simulink*. O operador poderá voltar ao controlo de posição largando o botão do punho, porém só poderá voltar a fazer controlo de admitância novamente após decorridos 2 segundos. Este tempo tem dois propósitos: inicializar o controlador de posição; assegurar que o operador sente que pode ter largado o botão inadvertidamente, impedindo assim que exista uma transição rápida de controlo que poderia causar efeitos indesejados.

3.3 Modelação do sistema físico e síntese de controladores

3.3.1 Modelação do sistema físico

O robô cartesiano é constituído por três eixos lineares com um sistema de transmissão mecânica. Cada eixo do robô pode ser caracterizado como um sistema de transmissão de fuso de esferas, composto por um acoplamento de rigidez e amortecimento próprio representado na Figura 3.3. O efeito das variações de inércia causadas por alterações na configuração da estrutura ou da carga manipulada, quando referidas aos motores pode ser desprezado, visto que tais variações aparecem divididas pelo quadrado do factor de redução (como já referido na descrição do robô cartesiano, secção 2.2, um valor elevado).

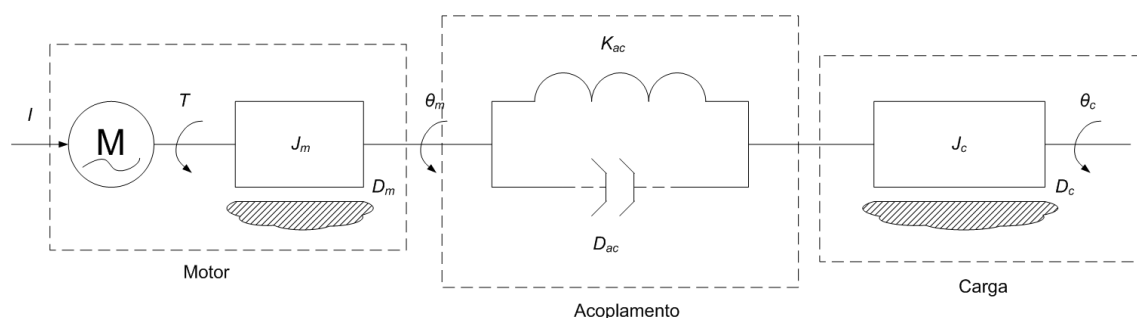


Figura 3.3 - Modelo de um sistema de transmissão com acoplamento de rigidez e amortecimento próprio.

É pois possível traduzir a dinâmica do sistema nas seguintes equações:

$$\begin{cases} T = K_t I \end{cases} \quad (3.1)$$

$$\begin{cases} T = J_m \ddot{\theta}_m + D_m \dot{\theta}_m + D_{ac}(\dot{\theta}_m - \dot{\theta}_c) + K_{ac}(\theta_m + \theta_c) \end{cases} \quad (3.2)$$

$$\begin{cases} J_c \ddot{\theta}_c + D_c \dot{\theta}_c + D_{ac}(\dot{\theta}_c - \dot{\theta}_m) + K_{ac}(\theta_c - \theta_m) = 0 \end{cases} \quad (3.3)$$

Na Tabela 3.3.1 estão enunciados os parâmetros usados utilizados na modelação do eixo linear.

Tabela 3.3.1 - Parâmetros utilizados na modelação de um sistema de transmissão com acoplamento de rigidez e amortecimento próprio.

Símbolo	Legenda (unidades S.I.)
T	Binário disponibilizado pelo motor (N.m)
K_t	Constante eléctrica do motor (N.m/A)
K_{ac}	Rigidez do acoplamento (N.m/rad)
I	Corrente eléctrica (A)
θ_m	Posição angular do motor (rad)
θ_c	Posição angular da carga (rad)
J_m	Inércia do motor (Kg.m ²)
J_c	Inércia da carga (Kg.m ²)
D_m	Atrito dinâmico do motor (N.m.s/rad)
D_c	Atrito dinâmico da carga (N.m.s/rad)
D_{ac}	Atrito dinâmico do acoplamento (N.m.s/rad)

A função de transferência do sistema é obtida a partir das equações (3.2) e (3.3), aplicando a *Transformada de Laplace* da seguinte forma:

$$\{ [J_m s^2 + (D_m + D_{ac})s + K_{ac}] \theta_m(s) - (D_{ac}s + K_{ac}) \theta_c(s) = T(s) \quad (3.4)$$

$$\{ [J_c s^2 + (D_c + D_{ac})s + K_{ac}] \theta_c(s) - (D_{ac}s + K_{ac}) \theta_m(s) = 0 \quad (3.5)$$

A partir da equação (3.5), obtém-se:

$$\frac{\theta_c(s)}{\theta_m(s)} = \frac{D_{ac}s + K_{ac}}{J_c s^2 + (D_c + D_{ac})s + K_{ac}} \quad (3.6)$$

Da manipulação das equações (3.4) e (3.5), resulta:

$$\frac{\theta_m(s)}{T(s)} = \frac{J_c s^2 + (D_c + D_{ac})s + K_{ac}}{s^4(J_c J_m) + s^3[J_c D_m + J_m D_c + (J_m + J_c)D_{ac}] + s^2[D_m D_c + D_{ac}(D_m + D_c) + K_{ac}(J_m + J_c)] + sK_{ac}(D_m + D_c)} \quad (3.7)$$

Sendo que a função de transferência final poder ser dada por:

$$\frac{\theta_c(s)}{T(s)} = \frac{\theta_c(s)}{\theta_m(s)} \frac{\theta_m(s)}{T(s)} \quad (3.8)$$

Logo,

$$\frac{\theta_c(s)}{T(s)} = \frac{D_{ac}s + K_{ac}}{s^4(J_c J_m) + s^3[J_c D_m + J_m D_c + (J_m + J_c)D_{ac}] + s^2[D_m D_c + D_{ac}(D_m + D_c) + K_{ac}(J_m + J_c)] + sK_{ac}(D_m + D_c)} \quad (3.9)$$

Desta forma o modelo em *Simulink* do eixo linear é apresentado na Figura 3.4:

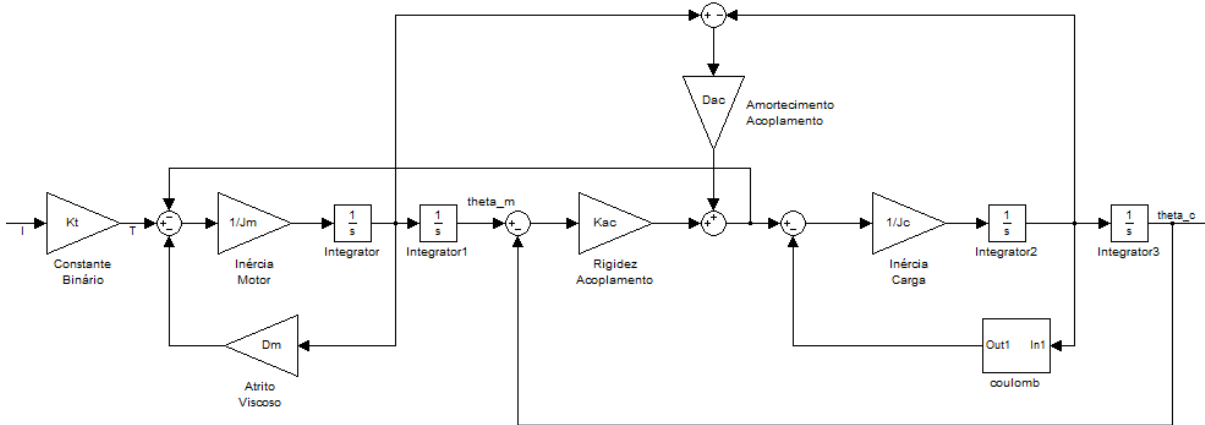


Figura 3.4 - Modelo em *Simulink* de um eixo linear.

O modelo apresentado pode ser simplificado considerando um acoplamento rígido (ideal) ($K_{ac} \rightarrow \infty$) e ($D_{ac} = 0$):

$$\frac{\theta_m(s)}{T(s)} = \frac{\theta_c(s)}{T(s)} = \frac{1}{s^2(J_c + J_m) + s(D_c + D_m)} \quad (3.10)$$

Sendo o modelo em *Simulink* correspondente apresentado na Figura 3.5:

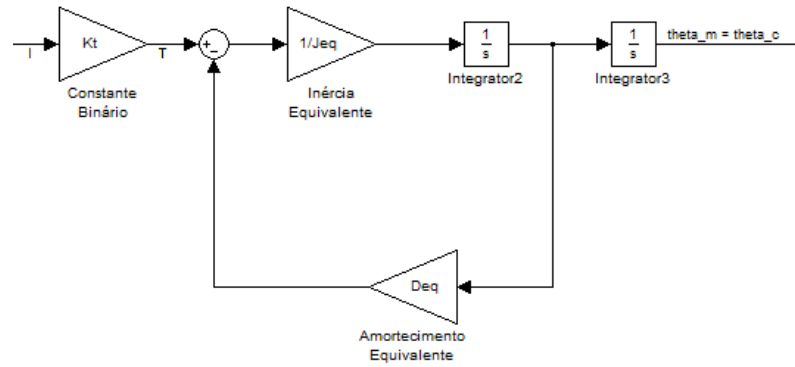


Figura 3.5 - Modelo simplificado em *Simulink* de um eixo linear.

No modelo *Simulink*, J_{eq} e D_{eq} representam a respectivamente a inércia e o atrito equivalentes, sendo dados por:

$$J_{eq} = J_c + J_m \quad (3.11)$$

$$D_{eq} = D_c + D_m \quad (3.12)$$

Para a modelação do sistema físico, foram utilizados os parâmetros descritos em (Cardoso, 2009) e que estão transcritos na Tabela 3.3.2.

Tabela 3.3.2 - Valores dos parâmetros que constituem o modelo.

Parâmetros	Valor (unidades S.I.)
K_t	0.590 N.m/A (fornecido pelo fabricante)
J_m	1.672E-4 Kg.m ² (calculado a partir dos valores fornecidos pelo fabricante)
J_c	eixo X -0.0024 Kg.m ² ; eixo Y – 0.0016 Kg.m ² ; eixo Z – 8.695E-4 Kg.m ² ; (calculado de acordo com os componentes montados)
D_m	0.001 N.m.s/rad (fornecido pelo fabricante)
D_c	0.001 N.m.s/rad (10% do binário do motor à velocidade máxima)
D_{ac}	4E-4 N.m.s/rad (valor estimado)
K_{ac}	12E3 N.m/rad (fornecido pelo fabricante)

3.3.2 Síntese de controladores

Os controladores a seguir apresentados estão associados aos estados da *Máquina de Estados*. Para cada estado foram implementados três controladores semelhantes, um por cada eixo. Assim, para no estado *Zero da Máquina* foram aplicados controladores de velocidade P e para os estados *Estabilizado* e *Trajectória Linear 3 eixos* foram aplicados controladores de posição PID. Estes controladores já foram alvo de estudo (Cardoso, 2009) pelo que apenas será feita uma breve descrição dos mesmos. Eles asseguram funções auxiliares no âmbito deste trabalho. A análise do seu desempenho foi, pelas mesmas razões descartada.

Para na *Manipulação Colaborativa* foram aplicados controladores de admitância. Estes controladores têm por base controladores de velocidade I-P, sendo esses também aqui descritos.

Controlador de velocidade PI

Os controladores de velocidade serão a base dos controladores de admitância para a manipulação colaborativa. Desse modo o controlador foi projectado para o robô fazer o seguimento dos movimentos impostos pelo operador. O sistema recebe uma referência em força e responde com a velocidade linear correspondente. Assim foram desenvolvidos, para cada um dos três eixos, controladores PI independentes.

Dado o modelo simplificado do eixo linear, a relação entre a corrente eléctrica e o binário disponibilizado pelo motor (3.1), e sabendo que o factor de conversão entre velocidade linear angular é dado pela relação de transmissão, $\dot{\theta} = K_{la} \dot{x}$, pode escrever-se:

$$\frac{sX(s)}{I(s)} = \frac{K_t}{sJ_{eq} + D_{eq}} \cdot \frac{1}{K_{la}} \quad (3.13)$$

$$\frac{sX(s)}{I(s)} = \frac{\frac{K_t}{D_{eq}K_{la}}}{s\frac{J_{eq}}{D_{eq}} + 1} \quad (3.14)$$

Simplificando a equação (3.14),

$$\frac{sX(s)}{I(s)} = \frac{K}{sT_m + 1} \quad (3.15)$$

O sistema que se pretende controlar é um sistema de primeira ordem, sendo:

$$K = \frac{K_t}{D_{eq}K_{la}} \quad (3.16)$$

$$T_m = \frac{J_{eq}}{D_{eq}} \quad (3.17)$$

Para assegurar erro nulo a referências de velocidade constantes e poder impor independentemente a frequência natural não amortecida do sistema, ω_n , e a razão de amortecimento, ζ , é necessário aumentar o tipo e a ordem do sistema controlado, assim sendo foi utilizado o controlador PI, $G_c(s)$:

$$G_c(s) \cdot G(s) = \left(K_p + \frac{K_i}{s}\right) \frac{K}{sT_m + 1} \quad (3.18)$$

Função de transferência em malha fechada obtida por manipulação da equação 3.18 vem:

$$G_{MF}(s) = \frac{\frac{(K_p s + K_i)K}{s(T_m s + 1)}}{1 + \frac{(K_p s + K_i)K}{s(T_m s + 1)}} = \frac{\frac{K_p}{K_i} s + 1}{s^2 \frac{T_m}{K_i K} + s \frac{1 + K_p K}{K_i K} + 1} = \frac{T_i s + 1}{\frac{s^2}{\omega_n^2} + \frac{2\zeta}{\omega_n} s + 1} \quad (3.19)$$

Em que,

$$\omega_n = \sqrt{\frac{K_i K}{T_m}} \quad (3.20)$$

$$\zeta = \frac{1 + K_p K}{2\sqrt{T_m K_i K}} \quad (3.21)$$

Como é possível observar na equação (3.19) podem ser definidos independentemente a frequência natural e a razão de amortecimento, sendo que,

$$T_i = \frac{K_p}{K_i} \quad (3.22)$$

Controlador de velocidade I-P

Para evitar uma excessiva sobre-elongação na resposta ao degrau a acção proporcional apenas é aplicada ao sinal de retroacção (controlador I-P). Os parâmetros são idênticos aos do controlador PI. A função de transferência vem dada pela equação (3.23):

$$G_{MF}(s) = \frac{1}{s^2 \frac{T_m}{K_i K} + s \frac{1 + K_p K}{K_i K} + 1} = \frac{1}{\frac{s^2}{\omega_n^2} + \frac{2\zeta}{\omega_n} s + 1} \quad (3.23)$$

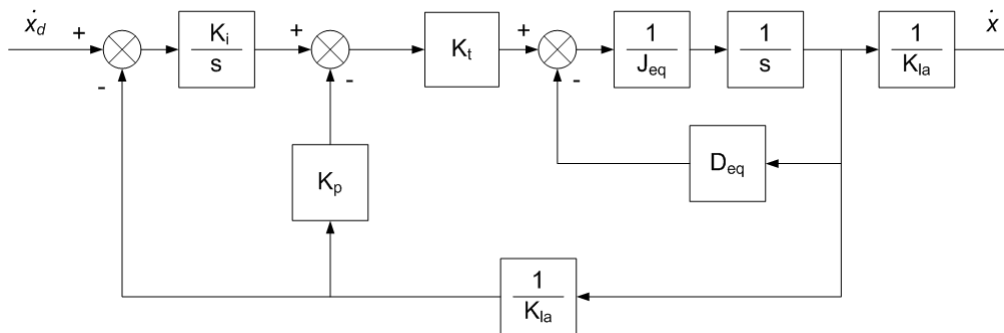


Figura 3.6 - Diagrama de blocos do controlador I-P de velocidade.

Este sistema representado na Figura 3.6 não apresenta zeros e segundo o critério de estabilidade de Routh (Ogata, 1997) é estável desde os seus coeficientes da equação característica sejam positivos.

A Figura 3.7 mostra o modelo do controlador definido em *Simulink* em tempo discreto:

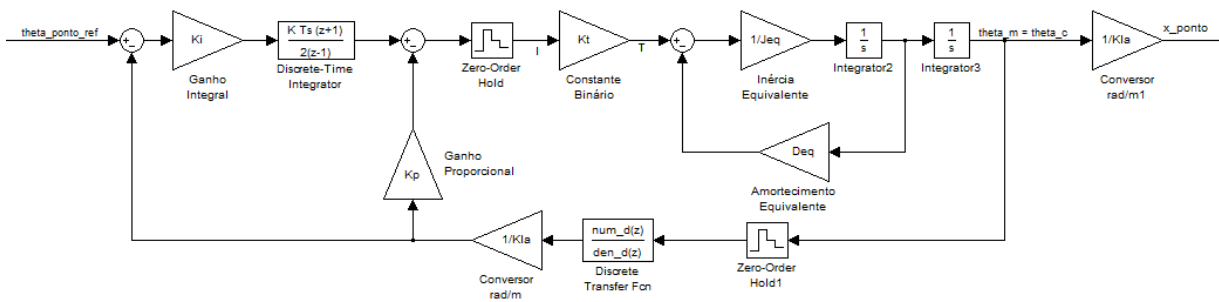


Figura 3.7 - Modelo em *Simulink* do controlador I-P aplicado ao modelo simplificado do eixo linear.

Como não se dispõe directamente do sinal de velocidade, aplica-se ao sinal de posição um filtro derivativo de primeira ordem, com uma frequência de corte de 100 Hz, com a seguinte função de transferência:

$$D(s) = \frac{As}{s + A} \quad (3.24)$$

Sendo $A = 2\pi \times 100$.

O controlador aqui apresentado foi modelado/simulado em tempo contínuo apesar de o sistema real funcionar em tempo discreto. Assim a escolha dos parâmetros do controlador ω_n e ζ necessita de ser alvo de um estudo cuidadoso.

O controlador em tempo discreto foi sintetizado por uma aproximação discreta do controlador contínuo. Esta abordagem obrigou à transformação dos parâmetros dos controladores para tempo discreto utilizando a *transformação bilinear* ou de *Tustin*. Desta forma todo o semi-plano negativo de *Laplace* é mapeado dentro do círculo unitário do plano z o que garante que a estabilidade do sistema é preservada. Porém a distorção frequencial obriga à utilização de frequências de amostragem elevadas.

A frequência natural do sistema foi escolhida tendo em conta o teorema da amostragem que refere que esta tem que ser maior ou igual que o dobro da maior frequência presente no sinal amostrado. A aplicação robótica, que funciona em tempo real, tem um tempo de amostragem (*step time*) de 1 ms. Porém o filtro derivativo para a aquisição da velocidade impõe frequências inferiores a 50 Hz, pois se o sistema for excitado a uma frequência superior o sinal de *feedback* de velocidade não representará a realidade de forma conveniente. Há ainda que incluir como factores para esta escolha o facto de existirem frequências não modeladas e de ter sido considerado um modelo físico de acoplamento ideal. O valor escolhido para a frequência natural foi 60 rad/s tendo por base as simulações e as experimentações envolvendo o controlo por admitância.

Para a escolha da razão de amortecimento, é necessário ter em consideração que se pretende ter uma resposta monótona do sistema controlado, visto que o efeito de *overshoot* é indesejado. Trata-se de uma aplicação em que operador está em contacto directo com o robô e como tal sente uma reacção deste sobre si próprio. Um exemplo indesejável da existência de *overshoot* é a incapacidade de parar o elemento terminal (velocidade nula) sem que o operador de aperceba que o elemento terminal mudou de direcção diversas vezes. Assim, de forma a garantir uma resposta monótona do sistema, escolheu-se $\zeta = 1$.

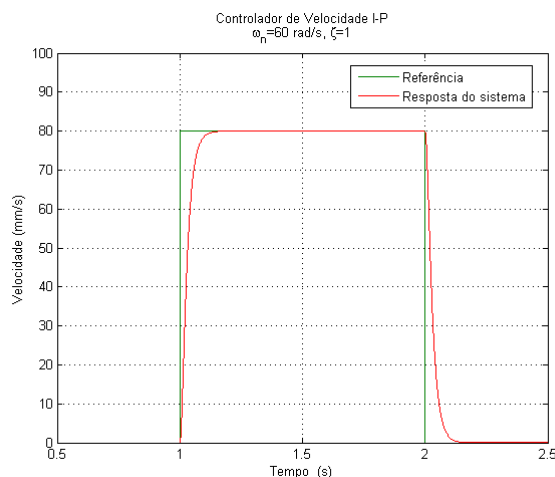


Figura 3.8 - Simulação da resposta do sistema ao degrau utilizando o modelo simplificado do eixo.

A Figura 3.8 mostra a resposta do sistema ao degrau. É possível observar que para os parâmetros do controlador escolhidos o sistema responde de forma monótona com um atraso de aproximadamente 0.2s.

A presença de uma acção integral no controlador e de saturação na resposta do actuador pode causar efeitos indesejados (*Windup*). O *Windup* do integrador é um efeito causado pelo aumento do valor do erro. Se o valor do erro aumentar ao ponto de o integrador saturar, a malha de realimentação deixa de responder. Mesmo que o erro diminua a acção de controlo permanecerá saturada devido ao facto do sistema estar a integrar valores muito elevados. Desta forma o sistema demorará um tempo demasiado longo a responder.

Para diminuir tal efeito podem ser utilizados vários métodos. Um deles impede o integrador de efectuar a integração quando o actuador se encontra num estado saturado. Um outro método corresponde a adicionar uma nova realimentação do erro, e_s , correspondente à diferença entre o sinal de saída do actuador, u_c , e a acção de controlo, v . Esse sinal é depois atenuado por um ganho $1/T_t$ que é adicionado ao sinal de entrada do integrador (Figura 3.9). Nos controladores implementados $T_t = T_i/2$ (de notar que T_t , constante de tempo, *tracking-time*, onde, por regra $T_t = T_i$, sendo que em alguns casos $T_t < T_i$ melhora a resposta do controlador).

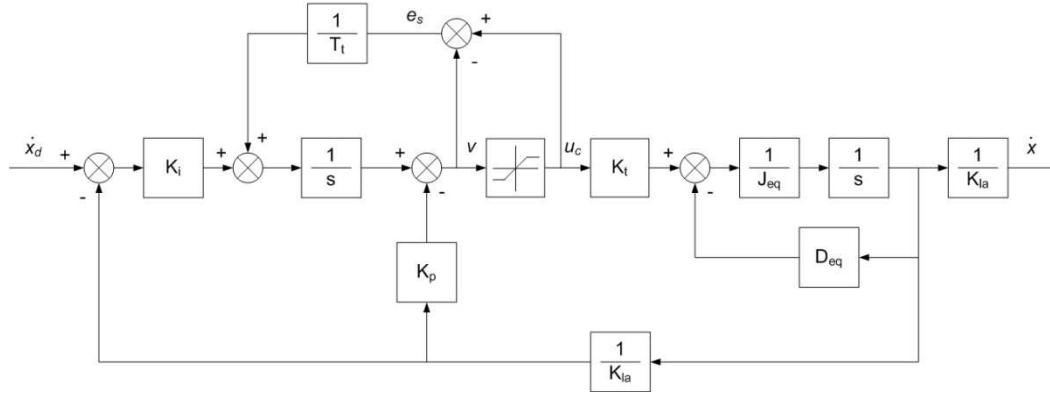


Figura 3.9 - Diagrama de blocos do controlador I-P de velocidade com malha anti-windup.

No sistema representado (Figura 3.9), o valor do erro e_s é nulo enquanto o actuador não estiver saturado. Sempre que o actuador satura, a realimentação tenta fazer com que o erro seja zero, efectuando o “reset” do integrador, de modo a que a acção de controlo se mantenha no limite da saturação (Aström, 1997). O sistema é pois tanto mais rápido a sair do estado saturado quanto menor for a constante de tempo T_t .

Controlador de admitância

O controlador de admitância é baseado no controlador de velocidade I-P sintetizado anteriormente. Este controlador permite ao operador movimentar directamente e sem esforço o elemento terminal em todo o espaço de trabalho.

Num eixo controlado por admitância a relação entre a força imposta pelo operador e o movimento real pode ser descrita como um atrito viscoso.

$$k sX(s) = F(s) \quad (3.25)$$

De forma equivalente,

$$sX(s) = \frac{1}{k} F(s) \quad (3.26)$$

Onde F é a força imposta pelo operador e $k > 0$ o amortecimento escolhido. Convenientemente define-se $c \equiv 1/k$, como sendo o ganho de admitância. Com esta definição compreende-se que para uma mesma força, o eixo reagirá com uma velocidade

tanto maior quanto maior for o ganho de admitância. Deste modo pode-se reescrever a função de transferência do controlador I-P descrita pela equação (3.23).

$$G_{MF}(s) = \frac{sX(s)}{F(s)} = \frac{c}{s^2 \frac{T_m}{K_i K} + s \frac{1 + K_p K}{K_i K} + 1} = \frac{c}{\frac{s^2}{\omega_n^2} + \frac{2\zeta}{\omega_n} s + 1} \quad (3.27)$$

Observa-se que c age como ganho em regime permanente não afectando o ajuste dos parâmetros do controlador de velocidade I-P.

A Figura 3.10 representa assim o modelo do controlador de admitância.

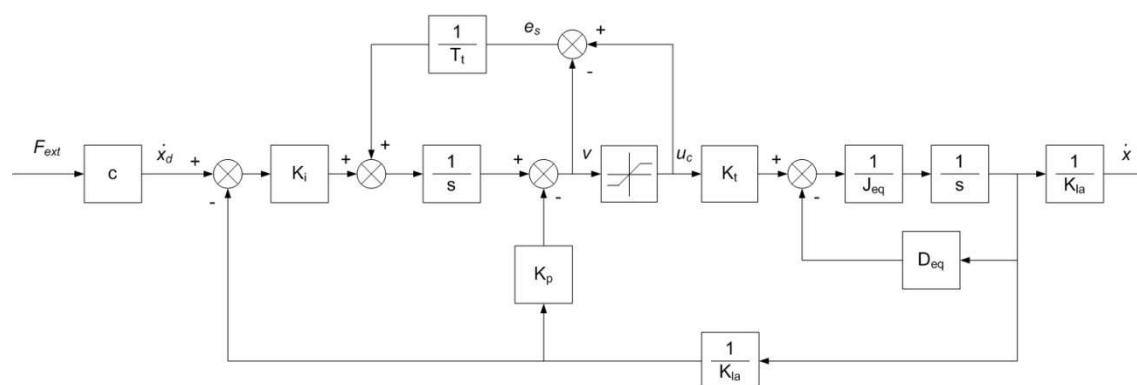


Figura 3.10 - Diagrama de blocos do controlador de admitância com malha anti-windup.

Controlador de velocidade P

A utilização de codificadores incrementais como transdutores de deslocamento, obriga a que sempre que é iniciado o robô seja efectuado o procedimento de inicialização dos codificadores (*Zero da Máquina*).

Como a informação de posição só está disponível após este procedimento, é inviável a utilização de um controlador PI (que requer uma derivação do sinal de posição) para os movimentos dados no procedimento *Zero da Máquina*. Assim para este procedimento foi utilizado um controlador de menor desempenho, controlador proporcional, que evidencia um erro não nulo a referências constantes.

Assim, este controlador pode ser representado pela seguinte expressão:

$$u(t) = K_p e(t) \quad (3.28)$$

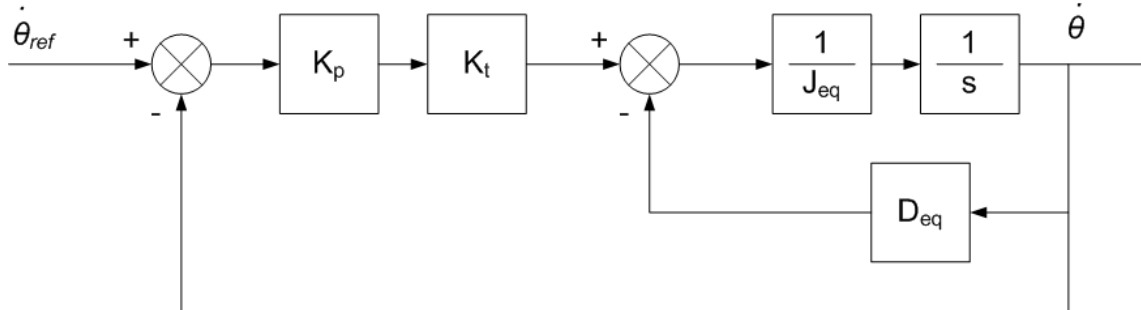


Figura 3.11 - Diagram de blocos do controlador P de velocidade.

Sendo que a função de transferência em malha fechada do sistema controlado representado na Figura 3.11 é a seguinte:

$$\frac{\dot{\theta}}{\dot{\theta}_{ref}} = \frac{K_p K_t}{J_{eq} s + K_p K_t + D_{eq}} \quad (3.29)$$

Os ganhos foram ajustados experimentalmente, tendo sido utilizados os seguintes valores para os eixos X, Y e Z: 0.694, 1 e 0.4, respectivamente.

Controlador de posição PID

Este controlador foi projectado para ser utilizado no seguimento de trajectórias de posição 3D, obtendo bons resultados no que diz respeito à capacidade de posicionamento. Ele está associado aos estados *Estabilizado* e *Trajectória Linear 3 eixos*. As trajectórias são definidas especificando as coordenadas dos pontos no espaço cartesiano e a velocidade média desejada. No estado *Estabilizado* as referências de posição dadas a este controlador são constantes e as referências de velocidade nulas de modo a manter o elemento terminal imobilizado (Figura 3.12).

Desta forma foram desenvolvidos controladores PID para cada um dos eixos (Cardoso, 2009).

Neste tipo de controladores, o modo integral é usado para eliminar o erro em regime estacionário causado por grandes variações de carga. O modo derivativo permite obter acções de controlo elevadas com o erro ainda pequeno, conduzindo ao aumento da velocidade de resposta do sistema.

A Figura 3.12 representa o modelo do controlador PID utilizado.

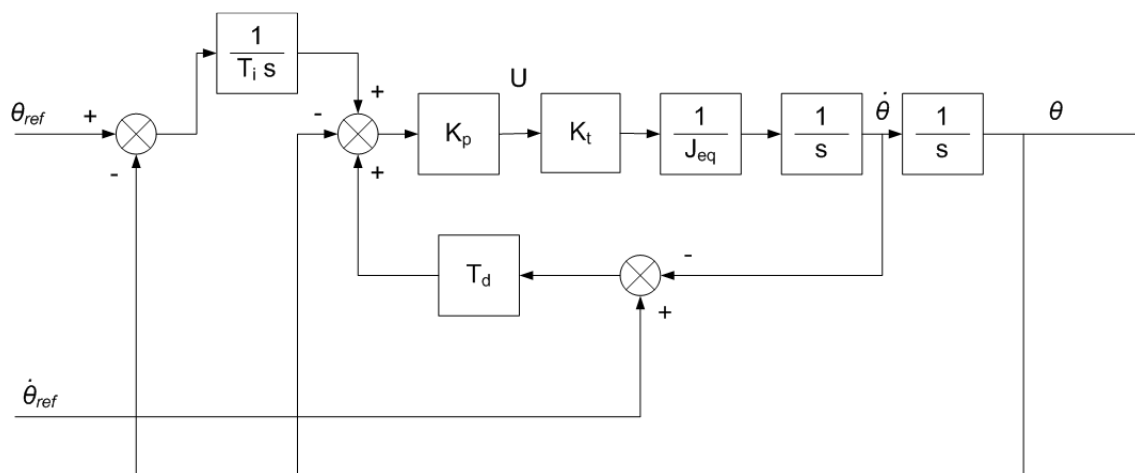


Figura 3.12 - Diagram de blocos do controlador PID de posição.

Foram ignorados tanto os atritos do sistema como o acoplamento.

A função de transferência deste sistema é:

$$\frac{\theta}{\theta_{ref}} = \frac{s^2 T_i T_d K_p K_t + K_p K_t}{s^3 J_{eq} T_i + s^2 T_i T_d K_p K_t + s T_i K_p K_t + K_p K_t} \quad (3.30)$$

Utilizando o *Protótipo de Bessel* (Ogata, 1997) foram determinados os valores dos parâmetros do controlador, sendo eles apresentados na Tabela 3.3.3:

Tabela 3.3.3 - Ganhos dos controladores de posição PID.

Ganho	Valor
K_{px}	30.387
T_{ix}	0.046

T_{dx}	0.018
K_{py}	21.480
T_{iy}	0.046
T_{dy}	0.018
K_{pz}	12.264
T_{iz}	0.046
T_{dz}	0.018

A obtenção da função de transferência, o ajuste dos ganhos dos controladores e a avaliação experimental do controlador PID de posição encontram-se descritos em (Cardoso, 2009).

3.3.3 Avaliação experimental do controlador I-P

Nesta secção será apresentado o resultado experimental da utilização do controlador de velocidade I-P. O controlador I-P foi implementado com base nas simulações em *Matlab/Simulink*. Os testes a este controlador consistiram na análise da resposta de um eixo a solicitações em degrau em espaço livre, tal como o representado na Figura 3.13 .

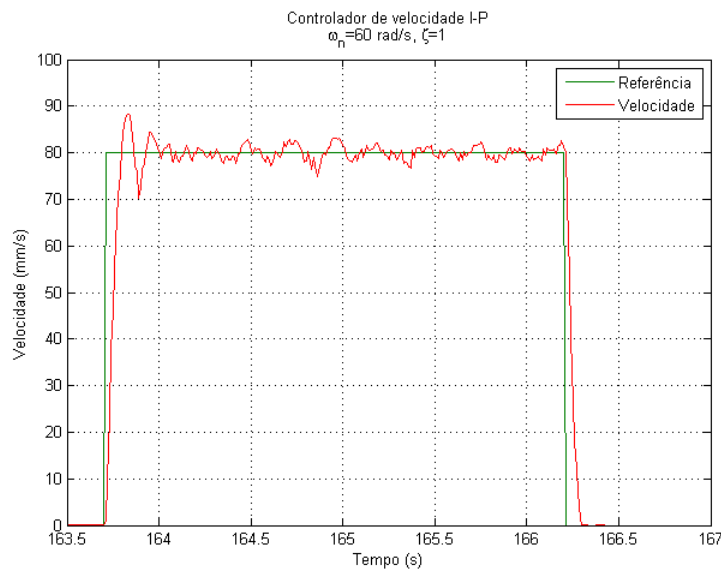


Figura 3.13 - Resposta do sistema real a uma solicitação em degrau em espaço livre.

Neste ensaio em espaço livre verificou-se que a resposta à solicitação em degrau (Figura 3.13) do eixo não apresenta a resposta monótona vista na simulação (Figura 3.8). Verificou-se ainda que o sistema apresenta alguma instabilidade quando solicitado por referências constantes. O efeito *overshoot* pode ser explicado por uma má modelação do sistema e pela sua simplificação a um eixo com acoplamento ideal. A instabilidade a referências constantes pode também ser explicada com a fraca qualidade do sinal de *feedback* de velocidade que é obtido por derivação da posição.

3.3.4 Avaliação experimental do controlador de admitância

Dada a natureza do sistema, onde o operador controla directamente o elemento terminal, era impossível que as solicitações do sistema fossem solicitações típicas (degrau, rampa, sinusoidal). Desta forma esta avaliação experimental é, pela natureza da aplicação, limitada, servindo apenas como forma de aferir a validade do controlador.

Também por essas razões foram tidos em conta outros parâmetros não mensuráveis como a melhor ou menor percepção do operador à resposta do sistema ou a facilidade de movimentação do elemento terminal.

Foram aferidos dados para diversos valores do ganho entre a força imposta pelo utilizador e a referência de velocidade dada aos controladores (ganho de admitância c).

Primeiramente foram efectuados ensaios com o operador a actuar sobre o elemento terminal do robô para um controlador de admitância de ganho, c unitário. A resposta do sistema obtida está representada na Figura 3.14 .

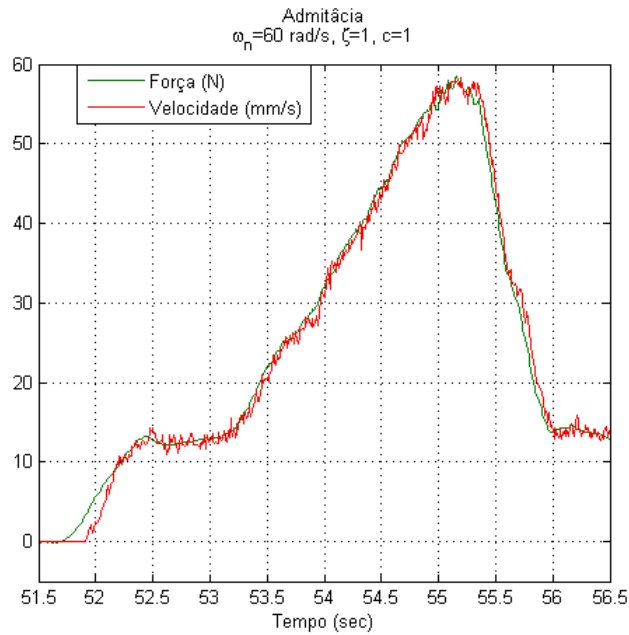


Figura 3.14 - Resposta do sistema a um controlador de admitância de ganho unitário.

Observa-se aqui também o ruído do sinal de velocidade (Figura 3.14). O sistema responde de acordo com a resposta ao degrau, evidenciando um atraso de sensivelmente 0.3s.

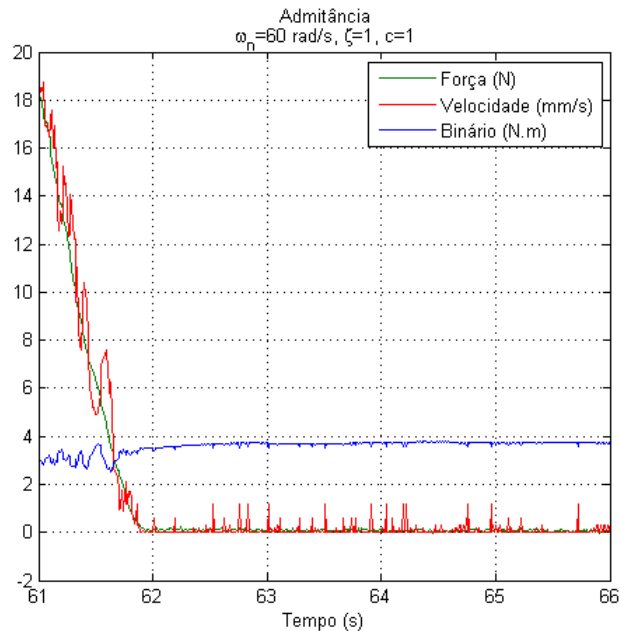


Figura 3.15 - Comportamento do sistema real para solicitações de velocidade reduzida.

No ensaio da Figura 3.15 é possível observar que o eixo só se move para binários aplicados pelos motores superiores a aproximadamente 3.5 N.m existindo assim uma zona

morta de resposta. Esta característica deve-se ao facto de se estar em presença de um sistema com elevada inércia e atrito (tanto estático como dinâmico).

Para o operador esta configuração do sistema, $c = 1$, é conservadora pois necessita de algum esforço para conseguir movimentar o elemento terminal com alguma agilidade. Não é perceptível ao operador o efeito de *overshoot* evidenciado na resposta ao degrau. Para tornar o sistema mais ágil foi aumentado o ganho do controlador de admitância, passando agora 1 N a corresponder a uma velocidade de 5 mm/s. O resultado do ensaio efectuado nas condições acima descritas é apresentado na Figura 3.16 .

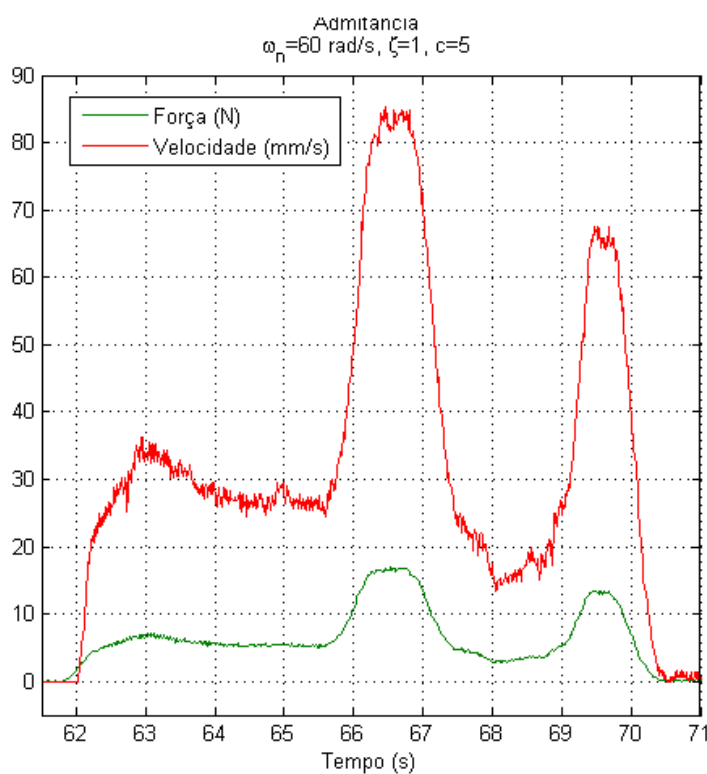


Figura 3.16 - Resposta do sistema real em controlo por admitância com ganho 5.

A alteração do ganho de admitância apenas se reflectiu num ganho de velocidade, sendo que a amplitude do ruído de velocidade se manteve constante o que corrobora a análise anteriormente feita de que o ruído se devia essencialmente à derivação do sinal de posição.

Para o operador esta configuração permite um menor esforço no manuseamento do elemento terminal, tendo uma resposta considerada mais agradável. Porém é agora mais perceptível o efeito *overshoot*, ainda que este não afecte o desempenho geral do manipulador.

Contudo, tratando-se de um controlo por admitância, o sistema apresenta problemas de instabilidade quando contacta com superfícies físicas rígidas. Um pequeno movimento do elemento terminal irá provocar um forte aumento nas forças de reacção. É o ambiente físico que fecha a malha de controlo, provocando um ganho elevado entre a velocidade e a força, levando à existência de instabilidade. Na síntese do controlador I-P tal efeito foi desprezado por se considerar que o braço/mão do operador não tinha rigidez suficiente para afectar o desempenho do sistema. Contudo para o ganho de admitância acima utilizado o sistema apresenta instabilidade se o operador movimentar o elemento terminal com o braço numa posição esticada, sendo essa a situação de contacto de maior rigidez imposta ao sistema. Esse efeito indesejado encontra-se representado na Figura 3.17 .

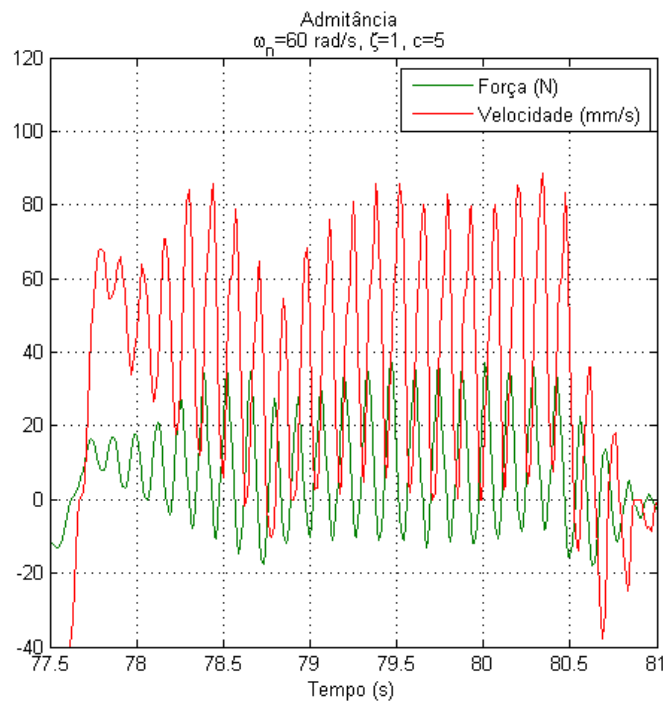


Figura 3.17 - Instabilidade do sistema real com aumento de rigidez do ambiente.

Uma das soluções para minimizar este problema foi o ajuste independente dos ganhos do controlador I-P. Para tal foi proposto reduzir o ganho integral, K_i . Esta solução permitia manter o ganho de admitância sem que se verificasse qualquer problema de instabilidade. O efeito negativo de tal ajuste foi o aumento do tempo de resposta do controlador. Verificou-se *a posteriori* que o atraso da resposta prejudicava a implementação das restrições virtuais pelo que esta solução foi abandonada.

A solução encontrada foi manter o ganho de admitância unitário. Não se verifica instabilidade embora o sistema robótico necessite de um considerável *input* de força para movimentar o elemento terminal de forma rápida.

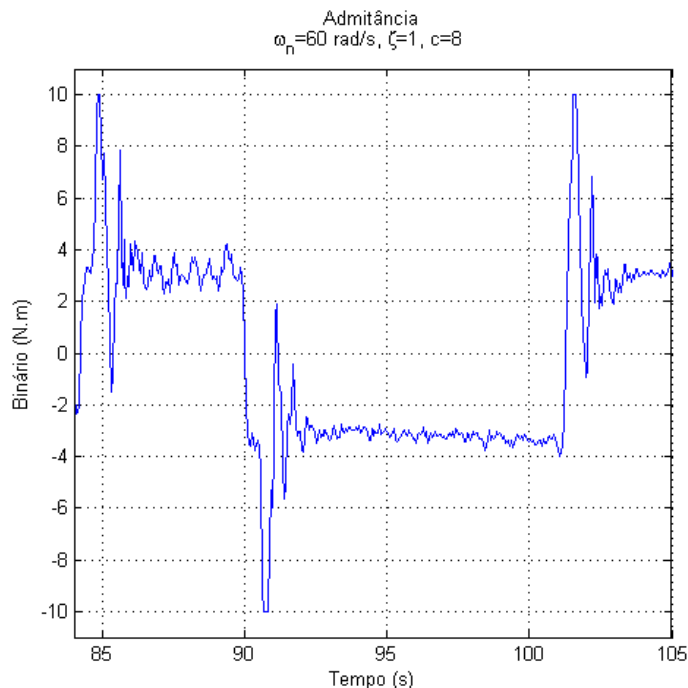


Figura 3.18 - Referência de binário do motor para elevadas solicitações de velocidade.

A Figura 3.18 representa a curva de resposta de binário do motor para uma solicitação elevada. É possível observar que apesar de o binário atingir o valor máximo o integrador do controlador não satura, permitindo assim retomar o controlo após essa situação crítica. Isso é possível graças à acção da malha *anti-windup*. Para este ensaio foi utilizado uma ganho de admitância 8 de modo a saturar mais facilmente o actuador.

3.4 Conclusões

O sistema robótico apresenta dois níveis de controlo. O primeiro, mais abrangente determina a resposta do sistema de acordo com o valor das receptividades num dado momento. É uma resposta de lógica sequencial de acordo com o especificado no diagrama de estado do sistema.

O segundo nível de controlo corresponde ao comportamento do sistema quando um determinado estado se encontra activo. Tal comportamento está associado ao bloco *Simulink* activado por cada estado. Deste modo, e de acordo com a tarefa pretendida, o sistema apresenta três controladores: controlador de velocidade P, que é usado para efectuar o *zero* dos *encoders*; controlador de posição PID, que é utilizado no posicionamento do elemento terminal num dado ponto do espaço de trabalho cumprindo uma determinada trajectória 3D; controlador de admitância/velocidade I-P, que permite o controlo directo do elemento terminal com recurso ao transdutor de força.

É possível concluir que a resposta do sistema real não é semelhante à do sistema simulado. Tentou-se reajustar os parâmetros determinados por simulação com o intuito de eliminar a instabilidade para ganhos de admitância diferentes de 1. Porém tal ajuste provocava um elevado atraso na resposta o que era incompatível com a implementação das restrições virtuais. Apesar disso o controlador real tem uma resposta satisfatória, sendo que o operador não sente qualquer instabilidade, atraso ou *overshoot* desde que mantida a configuração mais conservadora de admitância (ganho unitário).

Por último é possível concluir que apesar de ser extremamente difícil excitar o sistema no seu limite (situação de binário máximo), caso aconteça o sistema voltará, ao fim de breves instantes, ao seu controlo normal. Não existe assim uma saturação do integrador graças à presença da malha *anti-windup*.

Capítulo 4

Descrição e implementação das restrições virtuais

4.1 Introdução

As restrições virtuais impedem que um mecanismo se mova livremente num determinado volume dentro do seu espaço de trabalho. Deste modo as restrições virtuais só fazem sentido em aplicações colaborativas, onde a interacção física entre o operador e o robô permite um movimento conjunto. Este tipo de aplicações permitem que o operador mantenha o total controlo sobre o sistema (velocidade, posição, orientação) de uma forma bastante intuitiva e directa. É assim fácil perceber que o uso de restrições virtuais aumenta o desempenho do binómio Homem-Máquina em comparação com os sistemas que apresentam um movimento sem qualquer restrição. A restrição virtual é pois mais um instrumento de auxílio nas tarefas desempenhadas por estes sistemas.

O modelo de restrições descrito nas secções seguintes tem por base os estudos efectuados por Allison M. Okamura no protótipo *JHU Steady Hand Robot* (Bettini *et al.* , 2004). Este robô de 6 eixos, desenvolvido pela *The Johns Hopkins University*, é composto por eixos lineares e rotativos. O *JHU Steady Hand Robot* está representado na Figura 4.1.

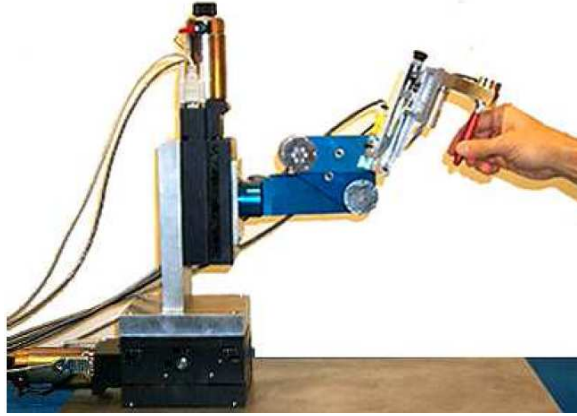


Figura 4.1 – O Steady Hand Robot.

Neste capítulo descrever-se-á a restrição virtual como sendo uma lei de controlo. Será dada ênfase a alguns casos tipo de aplicação como sendo as curvas, os tubos e os cones.

4.2 Restrição virtual como lei de controlo

Num sistema robótico colaborativo, a relação entre a força imposta pelo operador e o movimento real advém da condição de “contacto virtual” entre o elemento terminal e o ambiente. Em grande parte dos casos esse contacto é modelado por um atrito viscoso.

$$k \mathbf{v} = \mathbf{f} \quad (4.1)$$

De forma equivalente,

$$\mathbf{v} = \frac{1}{k} \mathbf{f} \quad (4.2)$$

Onde $\mathbf{f} \in \mathbb{R}^3$ é a força imposta pelo utilizador, $\mathbf{v} \in \mathbb{R}^3$ é a velocidade do robô, ambos vectores expressos no espaço cartesiano do robô. E onde $k > 0$ é o amortecimento do contacto virtual. Porém, será mais conveniente falar em termos de admitância, com $c \equiv 1/k$. Deste modo compreende-se que a velocidade do sistema numa dada direcção é proporcional à força exercida nessa mesma direcção, sendo essa resposta tanto maior quanto maior for a ganho de admitância c .

Observando a condição (4.2), conclui-se que o manipulador apresenta a mesma rigidez em qualquer direcção considerada, definindo-se assim como um sistema isotrópico controlado por admitância. Uma restrição virtual generaliza este modelo acrescentando uma ou mais condições de anisotropia, isto é, a rigidez será função da direcção em que se pretende mover o manipulador. Para isso considera-se a existência de uma matriz, $\boldsymbol{\delta} = \boldsymbol{\delta}(t)$, variável no tempo, de $3 \times n$, com $0 < n < 3$. De forma intuitiva, $\boldsymbol{\delta}$, representa as direcções “preferidas” de movimento do elemento terminal. Por exemplo, se $n = 1$, a direcção preferida é segundo uma curva no espaço, se $n = 2$, pretende-se executar movimentos numa superfície.

Usando a matriz $\boldsymbol{\delta}$, pode definir-se o operador projecção,

$$\mathbf{D}_\delta \equiv \boldsymbol{\delta} (\boldsymbol{\delta}^T \boldsymbol{\delta})^{-1} \boldsymbol{\delta}^T \quad (4.3)$$

A matriz $\boldsymbol{\delta}$ deve ser tal que $\boldsymbol{\delta}^T \boldsymbol{\delta}$ admita inversa.

Pode agora considerar-se a decomposição do vector força (exercida pelo operador) em duas componentes,

$$\mathbf{f}_\delta \equiv \mathbf{D}_\delta \mathbf{f} \quad (4.4)$$

$$\mathbf{f}_\tau \equiv \mathbf{f} - \mathbf{f}_\delta \quad (4.5)$$

Desta definição advém desde logo que $\mathbf{f}_\tau^T \mathbf{f}_\delta = 0$.

Combinando (4.4) e (4.2) resulta,

$$\mathbf{v} = c(\mathbf{f}_\delta + \mathbf{f}_\tau) \quad (4.6)$$

Pode agora introduzir-se uma nova admitância $c_\tau \in [0,1]$ que irá atenuar a resposta do sistema às componentes não preferidas da força, \mathbf{f}_τ . Daqui resulta o seguinte,

$$\mathbf{v} = c(\mathbf{f}_\delta + c_\tau \mathbf{f}_\tau) \quad (4.7)$$

$$\mathbf{v} = c(\mathbf{D}_\delta + c_\tau(\mathbf{I} - \mathbf{D}_\delta))\mathbf{f} = \mathbf{G}(c, c_\tau, \boldsymbol{\delta})\mathbf{f} \quad (4.8)$$

Esta última expressão é a forma geral de um controlo por admitância com os ganhos da matriz \mathbf{G} variáveis no tempo, dependendo de c , c_τ e $\boldsymbol{\delta}$. O valor c representa a admitância geral

do sistema. Impondo $c_\tau < 1$, é adicionada uma restrição virtual, dificultando a manipulação nas direcções ortogonais às direcções δ . No caso extremo de $c_\tau = 0$, tem-se uma restrição virtual rígida, visto que não é possível movimentar o elemento terminal em nenhuma outra direcção que não seja a direcção preferida. É ainda de notar que se $c_\tau = 1$, resulta um comportamento isotrópico.

4.3 Restrições virtuais para alvos e trajectórias

No caso de o sistema apresentar uma só direcção preferencial $n = 1$, podem ser distinguidos dois casos representados na Figura 4.2 :

- Ponto Alvo: a restrição virtual desejada é um segmento de recta no espaço tridimensional que liga em qualquer instante o ponto onde o elemento terminal se encontra e o ponto que se pretende atingir;
- Curva: a restrição virtual desejada é uma curva fixa no espaço cartesiano.

Para algumas tarefas estes dois tipos de comportamento podem ser combinados.

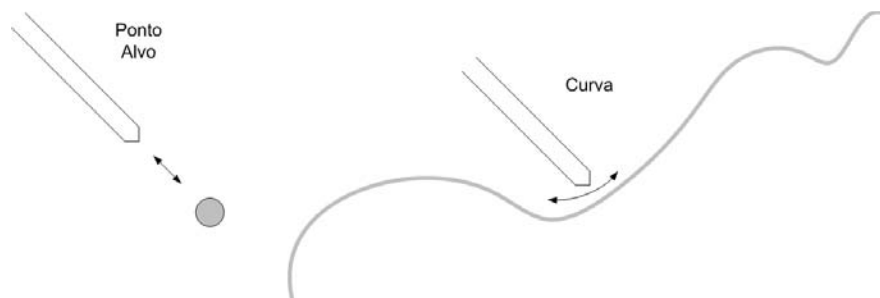


Figura 4.2 - Os dois tipos de comportamento que o sistema que utiliza restrições virtuais pode apresentar: Ponto Alvo e Curva.

Em ambos os casos é utilizada uma restrição virtual direccional que restringe o movimento a uma dada curva no espaço cartesiano. Para efeitos de controlo é assumido que a curva ou o alvo são previamente conhecidos. No caso do Ponto Alvo a restrição virtual é um segmento de recta que liga em cada instante o ponto alvo e a posição do elemento terminal.

4.3.1 Definição da direcção de referência

Considerando \mathbf{x}_a como a posição cartesiana do elemento terminal, a direcção de referência $\boldsymbol{\delta}$ é definida das seguintes formas:

- Caso 1: Ponto Alvo: a direcção de referência $\boldsymbol{\delta}_t$ é dada pela diferença entre a posição alvo $\mathbf{x}_t \in \mathbb{R}^3$ e a posição do manipulador:

$$\boldsymbol{\delta}_t(\mathbf{x}_a) = \mathbf{x}_t - \mathbf{x}_a \quad (4.9)$$

- Caso 2: Curva: assume-se que a restrição virtual é definida por uma expressão paramétrica:

$$\mathbf{p}(s) \equiv [x(s) \ y(s) \ z(s)]^T, \ s \in [0,1] \quad (4.10)$$

Definindo $\mathbf{p}(\hat{s}(\mathbf{x}_a))$ como sendo o ponto da curva mais perto do elemento terminal, segue-se:

$$\|\mathbf{p}(\hat{s}(\mathbf{x}_a)) - \mathbf{x}_a\| = \min_{s \in [0,1]} \|\mathbf{p}(s) - \mathbf{x}_a\| \quad (4.11)$$

Assim, a direcção de referência $\boldsymbol{\delta}_p$ é dada pelo vector normalizado tangente à curva nesse ponto.

$$\mathbf{t}(x_a) = \left. \frac{d}{ds} \mathbf{p}(s) \right|_{s=\hat{s}(\mathbf{x}_a)} \quad (4.12)$$

$$\boldsymbol{\delta}_p(\mathbf{x}_a) = \frac{\mathbf{t}(\mathbf{x}_a)}{\|\mathbf{t}(\mathbf{x}_a)\|} \quad (4.13)$$

Para o caso do Ponto Alvo é possível aplicar directamente a equação (4.8) com $\boldsymbol{\delta} = \boldsymbol{\delta}_t$. No entanto no caso Curva, definir $\boldsymbol{\delta} = \boldsymbol{\delta}_p$ só funcionaria como o esperado se o elemento terminal se mantivesse sobre a curva. Se a posição actual \mathbf{x}_a estiver fora do caminho seria imposto um movimento ao longo de uma curva paralela ao invés de voltar para o caminho desejado.

Para resolver este problema pode definir-se o erro cartesiano \mathbf{e} , como sendo a diferença entre o ponto da curva considerado e a posição do elemento terminal do robô.

$$\mathbf{e}(\mathbf{x}_a) = \mathbf{p}(\hat{s}(\mathbf{x}_a)) - \mathbf{x}_a \quad (4.14)$$

Sendo a nova direcção δ_c definida por,

$$\delta_c(\mathbf{x}_a) = \text{sin}(\mathbf{f} \cdot \delta_p(\mathbf{x}_a)) \delta_p(\mathbf{x}_a) + k_d \mathbf{e}(\mathbf{x}_a) \quad (4.15)$$

Onde o uso do operador *sin* garante a correcta convergência para a direcção δ . A maior ao menor resposta na convergência à curva é dada por $k_d > 0$, sendo que se este assumir valores elevados o sistema convergirá mais rapidamente mas terá uma menor margem de estabilidade.

4.3.2 Comportamento do sistema perto do ponto alvo

No caso de uma referência tipo Ponto Alvo, a aproximação ao objectivo leva à ausência de uma direcção de referência δ , impedindo assim o cálculo de \mathbf{D}_δ . Uma forma de contornar este problema é definir uma superfície esférica de comutação $S(\rho)$ de raio ρ em torno do ponto alvo. Assim, fora de $S(\rho)$, é aplicada a lei de controlo por admitância, enquanto que dentro da esfera é aplicada uma outra lei de controlo.

Uma das opções é corrigir a direcção de referência assim que se entra na esfera e depois utilizar essa referência para uma restrição virtual com uma direcção fixa. Esta solução pode levar a uma descontinuidade de velocidades na fronteira devido a uma mudança nas admitâncias.

A segunda opção é utilizar um controlo proporcional ao erro de posição dentro da esfera:

$$\mathbf{v} = k_c(\mathbf{x}_t - \mathbf{x}_a) \quad (4.16)$$

Em (4.16) k_c é um ganho escalar. Este controlador proporcional controla o elemento terminal ao longo de uma linha recta entre \mathbf{x}_a e \mathbf{x}_t .

4.4 Restrições virtuais para volumes

As anteriores definições de restrições têm por objectivo restringir o movimento a uma ínfima parte do espaço cartesiano, contudo em grande parte das aplicações o que se pretende é ter uma restrição do tipo volumétrica. Neste tipo de restrições pretende-se que o elemento terminal tenha um comportamento isotrópico dentro do volume e um comportamento anisotrópico fora desse volume. O comportamento anisotrópico assim criado deve obrigar o operador a dirigir o elemento terminal para dentro do volume. Uma vez dentro do volume o elemento terminal poderá movimentar-se igualmente em todas as direcções, mas será criada uma barreira que o impede de voltar ao ambiente anisotrópico.

4.4.1 Tubo

Pode supor-se agora que a tarefa pode ser descrita por uma curva paramétrica $\mathbf{p}(s)$ que representa o eixo de um tubo com raio r_t . A fronteira do tubo é uma superfície de comutação entre um movimento sem assistência (dentro do tubo) e uma restrição virtual atractora (fora do tubo). Mais, define-se uma região de transição ϵ com $0 < \epsilon < r$ dentro da qual o ganho c_t é suavizado. A restrição virtual do tipo Tubo está exemplificada na Figura (4.3).

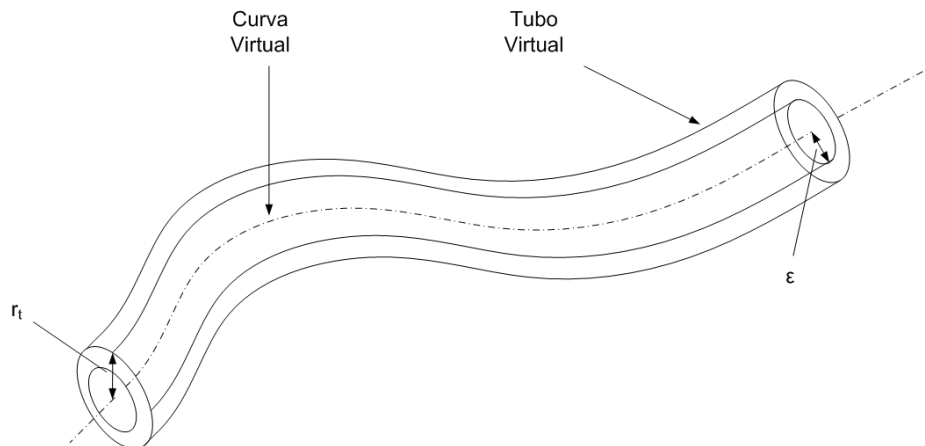


Figura 4.3 - Tubo virtual definido por uma curva paramétrica.

Pode fazer-se uso do erro cartesiano $\mathbf{e}(\mathbf{x}_a)$ definido na equação (4.14) e notar que o movimento é direccionado para fora do tubo quando $\mathbf{e}(\mathbf{x}_a) \cdot \mathbf{f} < 0$. Assim pode definir-se uma admitância ortogonal modificada c_{tu} da seguinte forma:

$$c_{tu}(\mathbf{e}) = \begin{cases} c_\tau, & \|\mathbf{e}\| > r_t \\ c_\tau - \left[\frac{r_t - \|\mathbf{e}\|}{\epsilon} \right]^n (c_\tau - 1), & (r_t - \epsilon < \|\mathbf{e}\| \leq r_t) \text{ e } (\mathbf{e} \cdot \mathbf{f} < 0) \\ 1, & \text{todos os outros casos} \end{cases} \quad (4.17)$$

$$(4.18)$$

$$(4.19)$$

Onde $n \geq 1$ é um valor escalar que modela a superfície de comutação, o caso particular $n = 1$ gera um função linear ao passo que um valor elevado cria uma região de comutação muito mais rígida. Isto é, o aumento de n leva a que $c_{tu}(\mathbf{e})$ seja mais próximo c_τ junto à superfície de comutação, desse modo é mais difícil a movimentação do elemento terminal para fora do volume definido.

O último passo é definir a direcção referência da restrição modificando δ_p de modo a obter a nova direcção δ_{tu} .

$$\delta_{tu}(\mathbf{x}_a) = \text{sin}(\mathbf{f} \cdot \delta_p(\mathbf{x}_a)) \delta_p(\mathbf{x}_a) + k_d \mathbf{e}_t(\mathbf{x}_a) \quad (4.20)$$

Onde,

$$\mathbf{e}_t(\mathbf{x}_a) = \begin{cases} 0, & \|\mathbf{e}(\mathbf{x}_a)\| \leq r_t \\ \frac{\mathbf{e}(\mathbf{x}_a)}{\|\mathbf{e}(\mathbf{x}_a)\|} (\|\mathbf{e}(\mathbf{x}_a)\| - r_t), & \|\mathbf{e}(\mathbf{x}_a)\| \geq r_t \end{cases} \quad (4.21)$$

$$(4.22)$$

Desta forma podemos aplicar a lei geral onde $\delta = \delta_{tu}(\mathbf{x}_a)$ e $c_\tau = c_{tu}(\mathbf{e})$.

4.5.1 Cone

A ideia de utilizar uma superfície cónica como restrição virtual é a de permitir guiar o movimento para um dado ponto A restrição virtual do tipo Cone está exemplificada na Figura 4.4 .

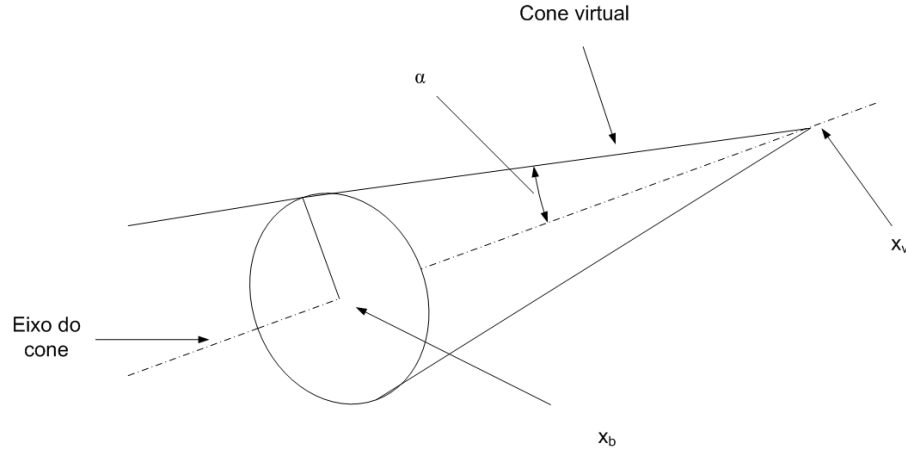


Figura 4.4 - Cone virtual definido pelo seu ponto alvo e pela abertura.

Para implementar um cone é necessário conhecer o seu vértice \mathbf{x}_v , o centro da base do cone \mathbf{x}_b e o ângulo de abertura α . A direcção do eixo do cone é então dada por,

$$\mathbf{a} = \frac{(\mathbf{x}_v - \mathbf{x}_b)}{\|\mathbf{x}_v - \mathbf{x}_b\|} \quad (4.23)$$

Dada a posição actual do elemento terminal no espaço cartesiano \mathbf{x}_a , é possível decompô-la nas suas componentes, ortogonais relativas ao eixo do cone, da seguinte forma:

$$\mathbf{n}_\perp(\mathbf{x}_a) = (\mathbf{I} - \mathbf{a} \cdot \mathbf{a}^T)(\mathbf{x}_v - \mathbf{x}_a) \quad (4.24)$$

$$\mathbf{n}_\parallel(\mathbf{x}_a) = (\mathbf{a} \cdot \mathbf{a}^T)(\mathbf{x}_v - \mathbf{x}_a). \quad (4.25)$$

Onde \mathbf{I} é uma matriz identidade 3×3 . Assim, define-se a distância do eixo do cone à fronteira como,

$$cone(\mathbf{x}_a) = \tan(\alpha) \|\mathbf{n}_\perp(\mathbf{x}_a)\| \quad (4.26)$$

Com todas estas definições, o elemento terminal do robô encontrar-se-á dentro do cone se a seguinte condição for satisfeita:

$$dentro(\mathbf{x}_a) \equiv \|\mathbf{n}_\perp(\mathbf{x}_a)\| \leq cone(\mathbf{x}_a) \quad (4.27)$$

Tal como no caso Tubo, pretende-se conseguir uma transição suave entre o interior e o exterior do cone. Assim, para cada posição \mathbf{x}_a dentro do cone (excepto os pontos sobre o

eixo), é possível calcular a direcção $\boldsymbol{\eta}$, normal à direcção $\boldsymbol{\delta}_t(\mathbf{x}_a)$, co-planar com $\mathbf{n}_\perp(\mathbf{x}_a)$, e a apontar para o interior do cone da seguinte forma:

$$\boldsymbol{\eta}(\mathbf{x}_a) = \boldsymbol{\delta}_t(\mathbf{x}_a) \times (\mathbf{n}_\perp(\mathbf{x}_a) \times \boldsymbol{\delta}_t(\mathbf{x}_a)) \quad (4.28)$$

Desta forma considera-se que o operador exerce força para fora do cone caso $\mathbf{f} \cdot \boldsymbol{\eta}(\mathbf{x}_a) \geq 0$. Da mesma forma que para o caso dos tubos pode definir-se uma zona de transição ϵ , sendo que o ponto \mathbf{x}_a se encontra dentro região se,

$$trans(\mathbf{x}_a) \equiv cone(\mathbf{x}_a) - \epsilon < \|\mathbf{n}_\perp(\mathbf{x}_a)\| \leq cone(\mathbf{x}_a) \quad (4.29)$$

Assim define-se o ganho de transição como,

$$\tilde{c}(\mathbf{x}_a) = c_\tau - \left[\frac{cone(\mathbf{x}_a) - \|\mathbf{n}_\perp(\mathbf{x}_a)\|}{\epsilon} \right]^n (c_\tau - 1) \quad (4.30)$$

Onde $n \geq 1$ é um valor escalar que modela a superfície de comutação.

No caso do cone, a zona de transição não pode vir definida por ϵ constante, isso seria impor que o elemento terminal só teria uma resposta isotrópica dentro de um cilindro com o eixo vertical na mesma direcção do eixo do cone e de raio ϵ . Assim utiliza-se ϵ_0 para definir o raio da base que terá uma resposta livre. O parâmetro ϵ é depois calculado a cada momento recorrendo à equação,

$$\epsilon = \left(\|\mathbf{n}_\perp(\mathbf{x}_a)\| - (\|\mathbf{x}_v - \mathbf{x}_b\| - \epsilon_0 \tan \alpha) \right) \tan \alpha \quad (4.31)$$

A Figura 4.5 explica melhor o comportamento do elemento terminal na restrição virtual cone.

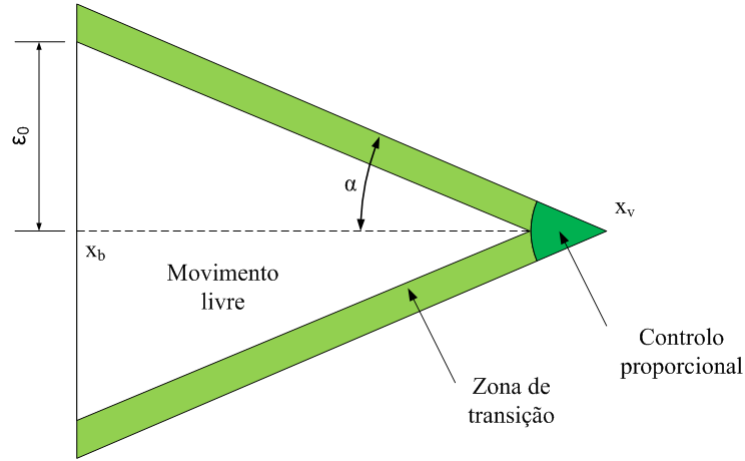


Figura 4.5 - Representação das diferentes zonas de admitância na restrição virtual do tipo Cone.

Finalmente definem-se os ganhos da função da seguinte forma,

$$c_{co}(\mathbf{x}_a) = \begin{cases} 1, & dentro(\mathbf{x}_a) \\ \tilde{c}(\mathbf{x}_a), & trans(\mathbf{x}_a) \text{ e } \mathbf{f} \cdot \boldsymbol{\eta}(\mathbf{x}_a) \geq 0 \\ c_\tau, & todos os outros casos \end{cases} \quad (4.32)$$

$$(4.33)$$

$$(4.34)$$

Tal como caso Tubo, quando o elemento terminal está fora do cone é criada uma admitância anisotrópica que promove o movimento no sentido do interior do cone (terceira condição). Quando o elemento terminal já se encontra dentro do cone é criada uma admitância isotrópica (primeira condição). Quando o elemento terminal se encontra dentro da zona de transição existe uma transição suave nos valores do ganho (segunda condição).

O último passo é definir a direcção referência da restrição modificando $\boldsymbol{\delta}_t(\mathbf{x}_a)$ de modo a obter a nova direcção $\boldsymbol{\delta}_{co}(\mathbf{x}_a)$.

$$\boldsymbol{\delta}_{co}(\mathbf{x}_a) = \text{sinal}(\mathbf{f}^T \boldsymbol{\delta}_t(\mathbf{x}_a)) \boldsymbol{\delta}_t(\mathbf{x}_a) + k_d \mathbf{e}_{co}(\mathbf{x}_a) \quad (4.35)$$

Em que o erro é dado por,

$$\mathbf{e}_{co}(\mathbf{x}_a) = \begin{cases} 0, & \|\mathbf{n}_\perp(\mathbf{x}_a)\| \leq cone(\mathbf{x}_a) \\ \frac{\mathbf{n}_\perp(\|\mathbf{n}_\perp(\mathbf{x}_a)\| - cone(\mathbf{x}_a))}{\|\mathbf{n}_\perp(\mathbf{x}_a)\|}, & todos os outros casos \end{cases} \quad (4.36)$$

$$(4.37)$$

Assim pode aplicar-se a equação geral de controlo (4.8) fazendo $\boldsymbol{\delta} = \boldsymbol{\delta}_{co}(\mathbf{x}_a)$ e $c_\tau = c_{co}(\mathbf{x}_a)$.

4.5 Implementação das restrições virtuais

As restrições virtuais são limitações impostas por software ao movimento dos eixos do sistema robótico. Assim, depois de implementados e testados os controladores de admitância (que permitem uma resposta livre em todo o espaço de trabalho), procedeu-se à implementação das restrições virtuais.

A abordagem face a este problema passou pela criação de um bloco *Simulink* que processasse a força que o utilizador impunha e gerasse referências de velocidade para os controladores de eixo, tendo em conta o tipo de restrição virtual e as suas características. Deste modo foi criada uma aplicação de controlo destinada a cada restrição virtual, sendo que a estrutura das aplicações (*Máquina de Estados*, *Trajectória Linear 3 eixos*, *Zero da Máquina*, etc.) se mantém igual, acrescentando-se apenas o bloco *Simulink* referente à restrição virtual.

O bloco *Simulink* da restrição virtual é activado a partir de um *output* da *Máquina de Estados* quando o estado *Manipulação Colaborativa* se encontra activo. O mesmo *output* activa também o bloco *Simulink* referente aos controladores de velocidade I-P. Dentro do bloco da restrição virtual encontra-se uma função nativa, *S-Function*, escrita em linguagem C e posteriormente compilada de forma a ser utilizada pelo *Matlab*. Nessa *S-Function* está escrita não só a lei geral de controlo por admitância como também as equações da restrição virtual e respectivos vectores tangentes no caso de se tratar de uma restrição do tipo Curva ou Tubo, por exemplo.

As *S-Function* poderiam ter sido escritas também em linguagem M, *M-file*. Essa abordagem diminuiria o tempo de desenvolvimento da aplicação pois a linguagem M é mais fácil de utilizar. Porém as *M-file* são muito restritas nos conteúdos, não permitindo, ao programador, o acesso directo a algumas funcionalidades. Outra desvantagem das *M-file* são os tempos de execução, maiores que as *S-Function* escritas em C.

Um ficheiro escrito em linguagem C para posterior compilação no *Matlab* tem que obedecer a uma estrutura fixa, denominada de *SimStruct*. Essa estrutura apresenta diversas rotinas (*callback methods*) com tarefas específicas, como por exemplo, a inicialização de portas (entradas e saídas da *S-Function*) e variáveis.

A parte inicial do código é composta por um cabeçalho que deve incluir o nível da *S-Function* e o seu nome, seguido pela declaração dos *header files* que se pretendem usar, sendo que o ficheiro *simstruc.h* é obrigatório pois é ele que dá acesso à estrutura da *SimStruct* e às funções *API* (*Application Program Interface*) do *Matlab* (The Math Works, 2002). A *simstruc.h* inclui já outros *header files* como por exemplo o *matrix.h* ou o *mex.h*. As variáveis globais e as funções escritas pelo programador devem ser declaradas logo após este cabeçalho e antes da rotina que define as características básicas da *S-Function* (*mdlInitializesizes*).

Depois de compilado o ficheiro C, este é executado pelo *Matlab* de acordo com uma sequência de rotinas representada na Figura 4.6. A execução do ficheiro compilado é dividida em três fases distintas. A inicialização, que inclui a execução das rotinas *mdlInitializesizes*, *mdlInitializeSampleTimes* e *mdlStart*, esta última nem sempre presente. Estas rotinas apenas são executadas uma vez logo no início da simulação. A segunda fase corresponde à execução da rotina *mdlOutputs* que é executada uma vez por cada ciclo de simulação. Os valores das saídas da *S-Function* serão recalculados a cada intervalo de amostragem até que a simulação termine. A última fase é a execução da rotina *mdlTerminate* que só acontece quando a simulação termina.

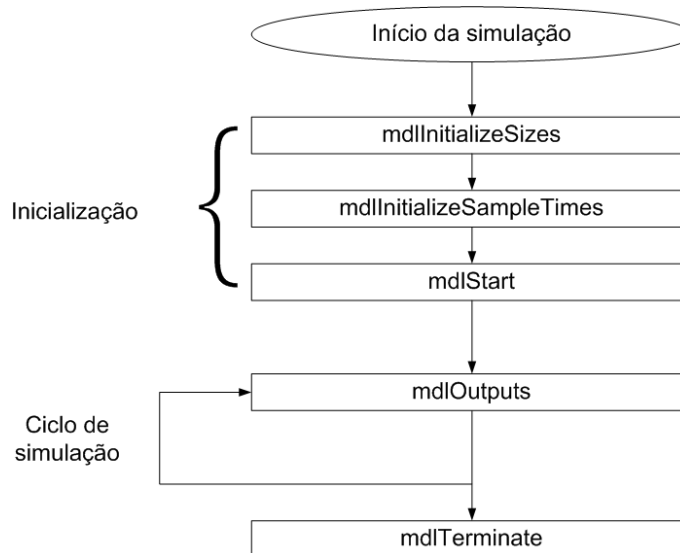


Figura 4.6 - Sequência de execução das rotinas da S-Function durante a simulação.

O *mdlInitializesizes* define as características básicas do bloco *S-Function* tais como o tempo de amostragem, o número de portas e a sua largura (variáveis simples ou indexadas).

O *mdlInitializeSampleTimes* define o tempo de amostragem e o tempo de *offset* de acordo com o tempo de amostragem do bloco *Simulink*.

O *mdlStart* corre apenas uma vez no início da simulação e permite executar operações que não devem ser executadas durante a simulação, como é o caso da geração da restrição virtual.

O *mdlOutputs* calcula as saídas da *S-Function* por cada intervalo de amostragem.

O *mdlTerminate* executa o fim das tarefas em execução.

O código das *S-Function* foi escrito de acordo com a restrição virtual em causa. Assim sendo para cada tipo de restrição virtual existe um código e uma estrutura diferente.

4.5.1 Ponto Alvo

Nesta restrição virtual o utilizador é assistido com o objectivo de levar o elemento terminal do robô a um determinado ponto do espaço de trabalho. Esta função deve ser assegurada independentemente da posição onde se encontra inicialmente o elemento terminal. O utilizador deve poder definir a admitância geral c , a admitância das direcções não preferidas c_t e o ponto do espaço de trabalho que se pretende atingir \mathbf{x}_t . O bloco *Simulink* correspondendo à *S-Function* implementada encontra-se representado na Figura 4.7 .

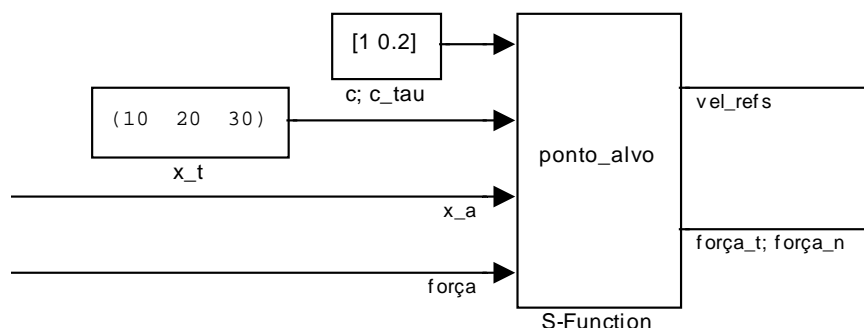


Figura 4.7 - Bloco *Simulink* que contém a restrição virtual Ponto Alvo.

Visto que não existe à partida uma trajectória definida não é necessário calcular os pontos e as direcções que o robô deve seguir durante o seu movimento. As direcções preferidas serão assim calculadas durante a execução do movimento, de acordo com a posição que o elemento terminal ocupa a cada intervalo de amostragem da *S-Function*. Deste modo a rotina *mdlStart* não é necessária.

A rotina *mdlOutput* contém o cálculo das direcções preferidas em cada intervalo de amostragem e a lei de controlo por admitância variável no tempo. Quando o elemento terminal se encontra a menos de 1mm (ajustado experimentalmente) do ponto alvo a *S-Function* aplica uma lei de controlo proporcional:

$$\mathbf{v} = k_c(\mathbf{x}_t - \mathbf{x}_a) \quad (4.38)$$

Esta, garante para valores pequenos de ganho k_c , a convergência do elemento terminal para o ponto pretendido.

Quanto o elemento terminal atinge o ponto permanece parado até que haja uma mudança de estado na *Máquina de Estados* (passagem para o estado *Estabilizado*).

Para validar a *S-Function* foi construído um modelo de simulação com 3 eixos baseados nos eixos reais. A admitância geral, $c = 1$ para todas as simulações efectuadas.

Foi primeiramente simulada uma situação em que o elemento terminal se move unicamente sobre um plano e cujos resultados se encontram representados na Figura 4.8. Esta situação permite uma melhor visualização da influência do parâmetro c_τ . Foram recolhidos dados relativos às forças tangenciais e normais (Figura 4.9) e às referências de velocidade (Figura 4.10).

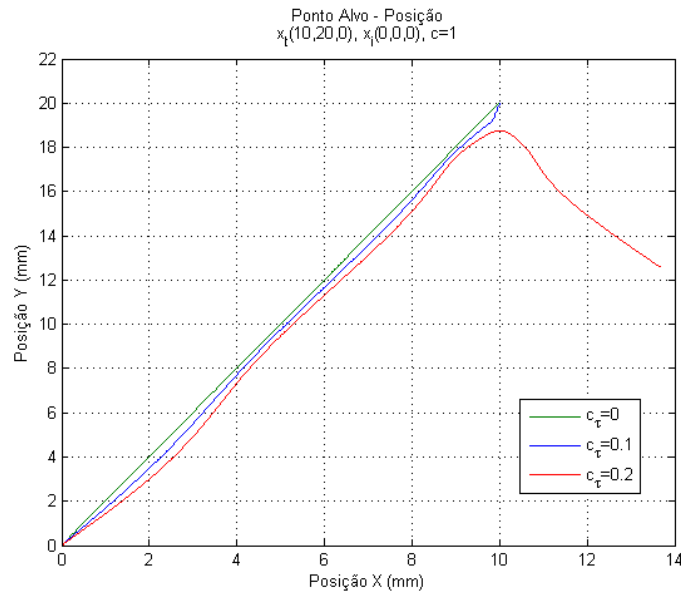


Figura 4.8 - Simulação do comportamento do elemento terminal para uma restrição do tipo Ponto Alvo.

É possível observar na Figura 4.8 que para $c_t = 0$ o sistema apenas deixa o operador movimentar o elemento terminal em direcção ao ponto alvo. O sistema permite movimentar o elemento terminal nos dois sentidos (aproximação e afastamento). O aumento do c_t leva a que o elemento terminal se possa mover noutras direcções que não em direcção ao ponto alvo. Como em cada novo ponto o sistema calcula uma nova direcção preferencial, a trajectória convergirá para o ponto para pequenos valores de c_t . Para elevados valores de c_t a convergência só é assegurada se para cada instante a força na direcção tangencial for maior que na direcção normal e de sinal tal que permita o movimento no sentido do ponto alvo.

Seguidamente são apresentadas as forças tangenciais e normais aplicadas no elemento terminal em relação à recta definida pelos pontos \mathbf{x}_t (ponto alvo) e \mathbf{x}_i (posição do elemento terminal no início da execução da restrição virtual), com $c_t = 0$ (Figura 4.9).

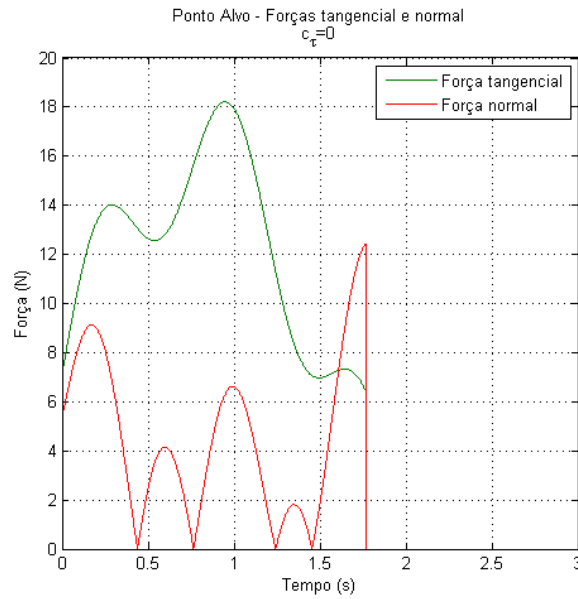


Figura 4.9 - Forças tangencial e normal no movimento utilizando restrição do tipo Ponto Alvo.

Os dados acima representados, Figura 4.9, permitem confirmar que apesar de existir uma componente de força normal à trajetória, o elemento terminal apenas se movimentou na direcção do ponto alvo.

Na Figura 4.10 são apresentadas as referências de velocidade geradas para os controladores de eixo.

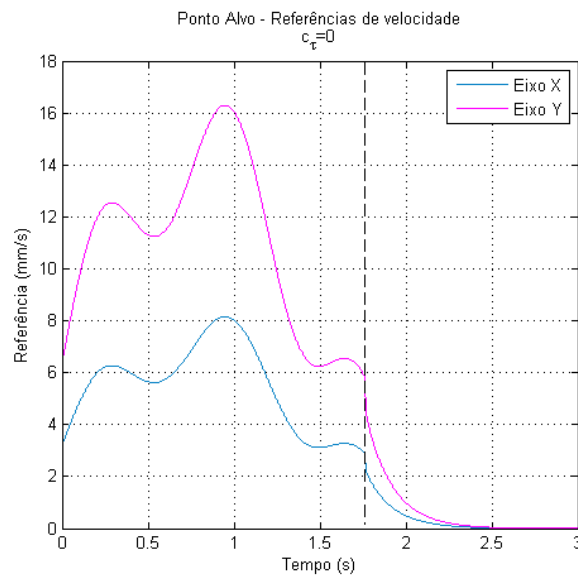


Figura 4.10 - Referências de velocidade dadas aos controladores de eixo para uma restrição virtual do tipo Ponto Alvo.

É possível observar que as referências dadas aos eixos, Figura 4.10, são proporcionais à componente tangencial da força, Figura 4.9 . Aos 1.75s o elemento terminal entra na esfera de raio 1mm e centro em \mathbf{x}_t . A partir desse momento é aplicada a lei de controlo proporcional em função do erro $\mathbf{x}_t - \mathbf{x}_a$. A maior ou menor descontinuidade que se poderá observar na passagem para o controlador proporcional está dependente do ganho proporcional k_c e da intensidade da força tangencial nesse momento.

Foi depois simulada uma situação em que o elemento terminal se move no espaço para corroborar os dados registados na situação planar. Os resultados dessas simulações estão apresentados na Figura 4.11 .

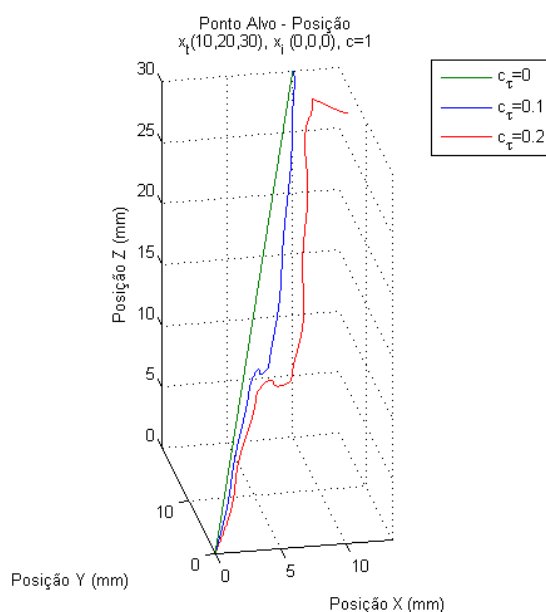


Figura 4.11 – Simulação do comportamento do elemento terminal nos três eixos cartesianos para uma restrição do tipo Ponto Alvo.

Verifica-se que a análise feita para o caso de uma movimentação planar pode ser transposta para uma movimentação no espaço.

4.5.2 Curva

Nesta restrição virtual o utilizador é assistido com o objectivo de o levar o elemento terminal do robô a percorrer uma determinada curva no espaço de trabalho. Caso o utilizador inicie a tarefa com o elemento terminal não localizado sobre a curva o sistema assiste o

utilizador num movimento de aproximação ao ponto da curva que em cada momento se encontra mais próximo. Neste trabalho é abordada uma curva helicoidal com o eixo vertical segundo a direcção Z. Esta curva foi escolhida por ser uma curva contínua, apresentar uma parametrização de fácil implementação e ter derivadas contínuas.

O utilizador pode definir os seguintes parâmetros da curva: centro da curva helicoidal \mathbf{x}_c ; raio da helicoidal r ; parâmetro passo da helicoidal b , admitância geral c ; admitância nas direcções não preferenciais c_τ (Figura 4.12).

As equações paramétricas da curva utilizada são:

$$x(s) = x_c + r \cos(2\pi s) \quad (4.39)$$

$$y(s) = y_c + r \sin(2\pi s) \quad (4.40)$$

$$z(s) = z_c - 2\pi b s \quad (4.41)$$

Com $s \in [0,1[$ em incrementos de 0.0001 de forma a gerar 10.000 pontos necessários para curva. De igual modo, as derivadas em cada ponto são dadas por:

$$\dot{x}(s) = -2\pi r \sin(2\pi s) \quad (4.42)$$

$$\dot{y}(s) = 2\pi r \cos(2\pi s) \quad (4.43)$$

$$\dot{z}(s) = -2\pi b \quad (4.44)$$

A direcção $\delta(s)$ em cada ponto s é dada:

$$\delta(s) = \frac{[\dot{x}(s) \ \dot{y}(s) \ \dot{z}(s)]^T}{\|[\dot{x}(s) \ \dot{y}(s) \ \dot{z}(s)]^T\|} \quad (4.45)$$

A Figura 4.12 apresenta o bloco *Simulink* da *S-Function* utilizada na implementação da restrição virtual do tipo Curva.

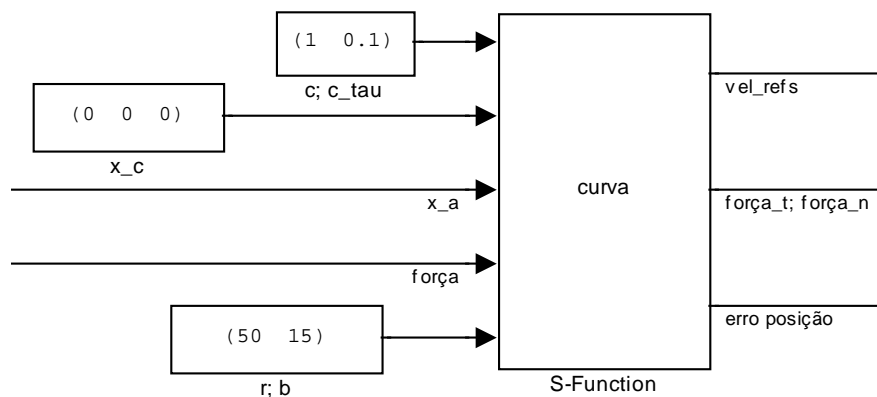


Figura 4.12 - Bloco Simulink que contém a restrição virtual do tipo Curva.

Na rotina *mdlStart* são calculados os pontos que definem a curva e a respectiva direcção preferencial. Estes valores são guardados em variáveis indexadas globais que são posteriormente utilizadas na rotina *mdlOutputs*. Esta abordagem permite calcular estes valores apenas uma vez, exigindo assim menos capacidade de cálculo por cada ciclo de simulação da *S-Function*. Nesta restrição virtual são calculados 10.000 pontos por curva.

A rotina *mdlOutput* contém a lei de controlo por admitância variável no tempo.

No início da tarefa o elemento terminal pode não se encontrar sobre a curva. É necessário corrigir δ de modo a que o elemento terminal se aproxime da curva (4.14) (4.15). Para tal é necessário saber qual o ponto da curva mais perto do ponto robô \mathbf{x}_a em cada ciclo de simulação da *S-Function* (4.11). Existiam inicialmente duas abordagens a este problema: procurar o ponto mais próximo dentro de um intervalo de pontos relativamente pequeno (mais/menos 50 pontos relativamente ao último ponto utilizado); procurar o ponto mais próximo varrendo todos os pontos da curva.

A primeira abordagem exige menos esforço de computação, pois só é necessário percorrer 100 pontos em 10.000 existentes. Porém esta abordagem necessita de uma inicialização (indicação de um ponto para início da busca). Tal inicialização está incluída na rotina *mdlStart*. Um outro inconveniente é a necessidade de ajuste do intervalo em função da admitância geral. Este ajuste evita que o ponto actual \mathbf{x}_a esteja já fora do intervalo de pontos definidos no ciclo de simulação anterior.

A segunda abordagem é percorrer em cada ciclo de simulação toda a curva e encontrar o ponto mais próximo do elemento terminal. Esta abordagem não necessita de inicialização. Porém também ela apresenta desvantagens, aumentando o esforço de cálculo necessário. Uma outra desvantagem é que caso a curva passe mais do que uma vez sobre o mesmo ponto existe mais do que uma direcção δ nesse ponto, podendo levar ao erro do operador. Depois de analisados os dados relativos aos tempos totais de execução, verificou-se que mesmo com 10.000 pontos por curva a aplicação não atingia a tempo limite de ciclo, 1ms, pelo que se optou por implementar esta abordagem. O código em linguagem C relativo a esta abordagem pode ser visualizado na Figura 4.13 .

```
//procura pelo ponto pertencente à curva mais perto do elemento
terminal
dmin=100000;
sm=0;
for (i=0;i<=dim-1;i++){

    di=sqrt(((traje_x[i]-(*uPtrs2[0]))*(traje_x[i]-
(*uPtrs2[0])))+((traje_y[i]-(*uPtrs2[1]))*(traje_y[i]-
(*uPtrs2[1])))+((traje_z[i]-(*uPtrs2[2]))*(traje_z[i]-(*uPtrs2[2]))));

    if (di<dmin){
        dmin=di;
        sm=i;
    }
}
```

Figura 4.13 - Exemplo de código em linguagem C para encontrar o ponto pertencente à restrição virtual mais próximo do elemento terminal.

Se o elemento terminal se encontrar a menos de 1mm dos pontos extremos da curva, aplica-se uma lei de controlo proporcional de modo a que o elemento terminal atinja de forma segura esses pontos.

Tal como no caso da restrição virtual do tipo ponto alvo, também para esta restrição foram efectuadas simulações para diferentes valores de c_τ (Figura 4.14).

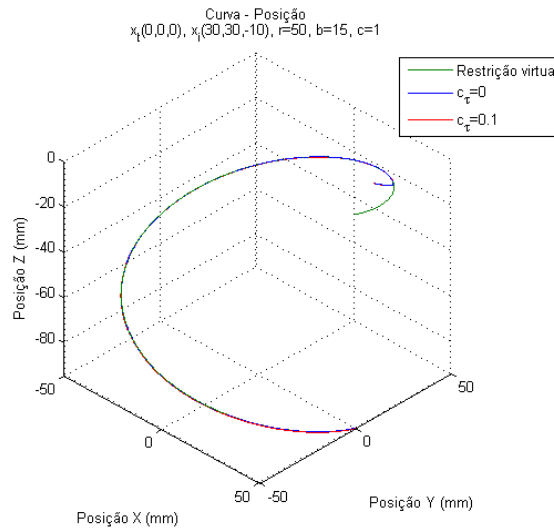


Figura 4.14 - Simulação do comportamento do elemento terminal para uma restrição virtual do tipo Curva.

É possível observar que mesmo iniciando o movimento fora da restrição $\mathbf{x}_i(30,30,-10)$ (posição do elemento terminal no início da execução da restrição virtual), o elemento terminal rapidamente converge para a curva.

Apesar de à primeira vista parecer que nos dois os casos o elemento terminal percorre a curva com erro nulo, ou próximo disso, a verdade é que isso só acontece para $c_t = 0$. Tal facto será clarificado mais à frente nesta análise às simulações efectuadas (Figuras 4.15 e 4.17).

Na Figura 4.15 é apresentado o erro do elemento terminal face ao ponto da curva mais próximo em cada momento.

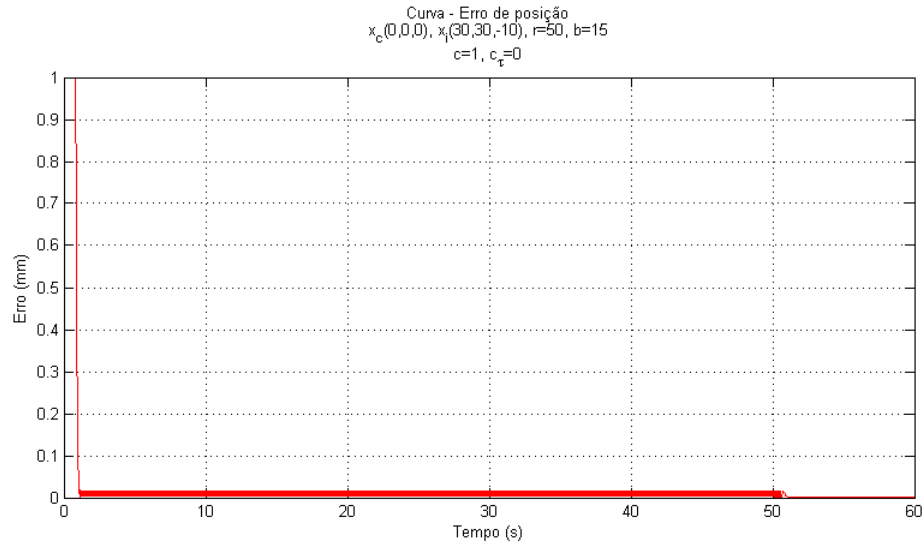


Figura 4.15 - Erro de posição em relação à restrição virtual para uma restrição do tipo Curva.

Observa-se que para $c_t = 0$ o elemento terminal aproxima-se da restrição e segue sobre ela com erro praticamente nulo. O erro apresentado a partir do momento em que o elemento terminal atinge a restrição é devido à distância entre os pontos que definem a curva e à dinâmica do sistema. Compreensivelmente, quanto mais espaçados se apresentarem os pontos que definem a restrição virtual maior será o erro no seguimento da restrição virtual.

Na Figura 4.16 são apresentados os dados relativos às componentes tangenciais e normais da força aplicada no elemento terminal. Considera-se como direcção tangencial a direcção preferencial δ do ponto da curva que, em cada instante, se encontra mais próximo do elemento terminal (4.11).

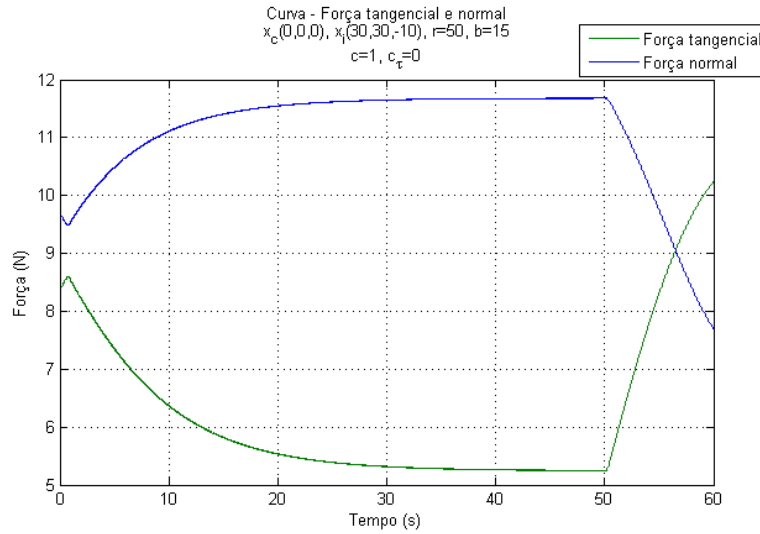


Figura 4.16 - Forças tangencial e normal no movimento utilizando restrição do tipo Curva.

Observa-se que mesmo sendo a componente normal maior que a componente tangencial o elemento terminal segue sempre sobre a restrição.

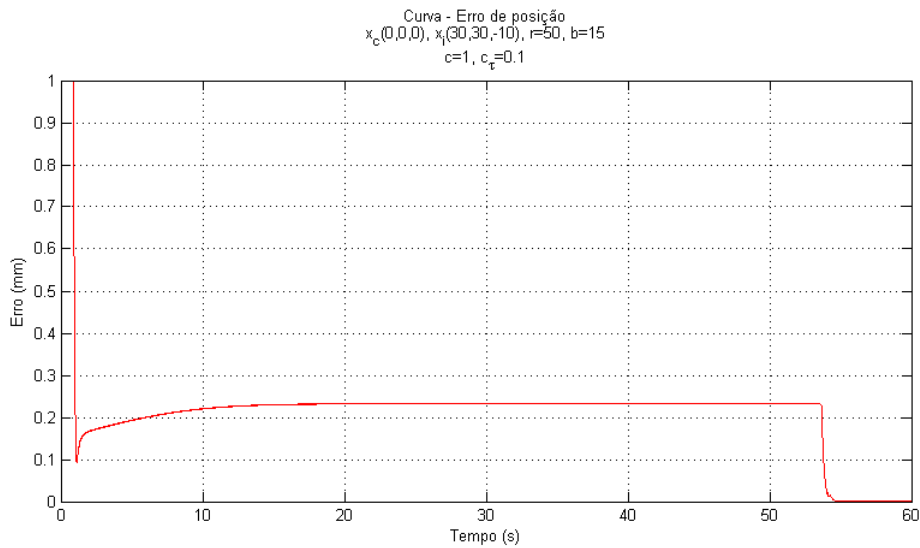


Figura 4.17 – Erro de posição em relação à restrição virtual para uma restrição do tipo Curva para $c_t = 0.1$.

Para $c_t = 0.1$ o erro de posição relativamente ao ponto mais próximo da curva (Figura 4.17) não é nulo. Na parte final da simulação o elemento terminal aproxima-se de um extremo da restrição e transita-se para a lei de controlo proporcional.

Verificou-se ainda que para valores de c_τ superiores a 0.1, nas mesmas condições de teste, o elemento terminal não cumpria, como é natural, a restrição virtual, pelo que tais resultados não foram incluídos nesta análise.

4.5.3 Tubo

Na restrição virtual do tipo tubo o utilizador é levado a movimentar o elemento terminal do robô dentro de um volume de secção constante, sendo que dentro desse volume o movimento é completamente livre. Se o elemento terminal se encontrar fora desse volume o utilizador é assistido a dirigir-se para dentro do mesmo. É utilizada uma curva helicoidal pelas mesmas razões que na restrição virtual do tipo Curva.

As equações utilizadas para a definição do perfil do tubo são as mesmas que para a restrição virtual do tipo Curva.

O utilizador poderá definir o raio da secção r_t , a região de transição ϵ e o parâmetro que modela a superfície de transição n , para além dos parâmetros já referidos na restrição virtual do tipo Curva. O bloco *Simulink* da *S-Function* que implementa a restrição virtual do tipo Tubo está representado na Figura 4.18.

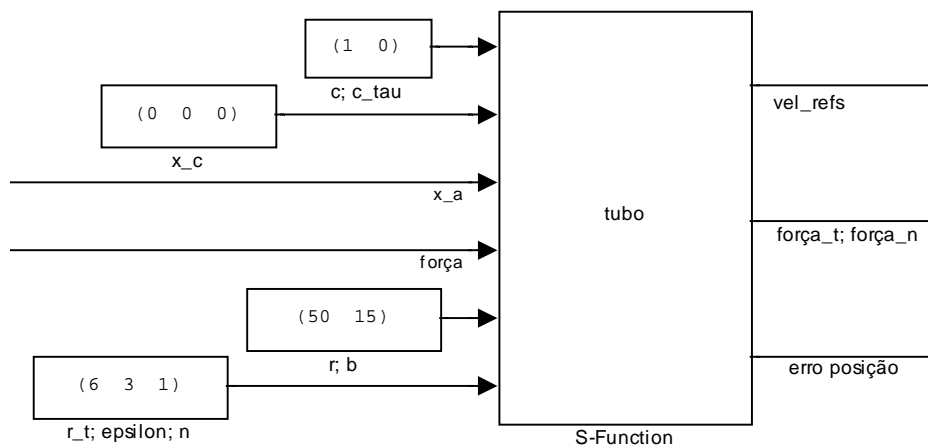


Figura 4.18 – Bloco Simulink que contém a restrição virtual do tipo Tubo.

A rotina *mdlStart* é semelhante à mesma rotina da restrição virtual do tipo Curva.

Na rotina *mdlOutputs* é calculada a lei de controlo por admitância variável de acordo com o comportamento pretendido. Tal como na restrição virtual do tipo Curva também aqui foi implementado o código que permite a procura, em cada momento, do ponto da curva mais próximo do elemento terminal.

Se o utilizador se encontrar junto aos extremos da curva a *S-Function* aplica de forma automática a convergência para os respectivos pontos através de uma lei de controlo proporcional.

Esta restrição virtual foi também alvo de simulações. Foi simulada a situação em que inicialmente o elemento terminal se encontra fora do tubo e que se encontra apresentada na Figura 4.19.

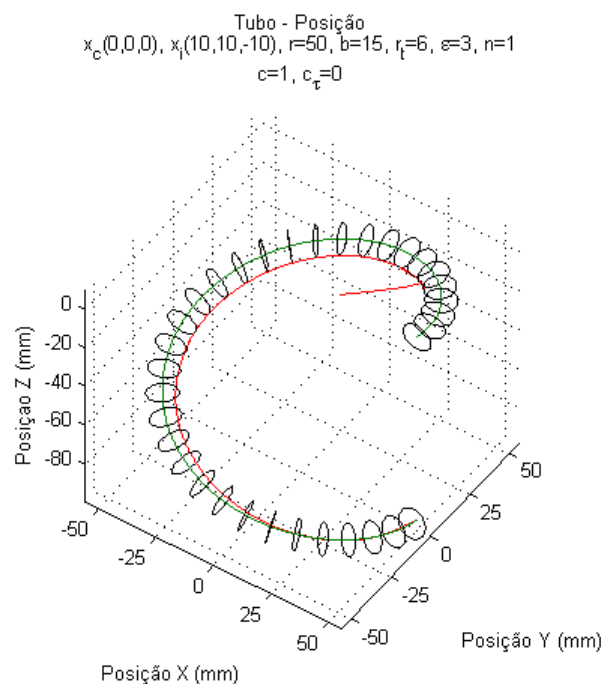


Figura 4.19 - Simulação do comportamento do elemento terminal para uma restrição virtual do tipo Tubo.

É possível observar (a vermelho) o movimento efectuado pelo elemento terminal. A linha a verde representa a curva que define o tubo. Verifica-se que inicialmente o elemento terminal aproxima-se do tubo e de seguida segue junto no interior deste até atingir o extremo. O facto de, depois entrar no tubo, o elemento terminal se mover junto à parede do tubo deve-se à dificuldade em ajustar as condições de simulação.

Quando observado na direcção do eixo Z, torna-se evidente que o elemento terminal não abandona o tubo (Figura 4.20).

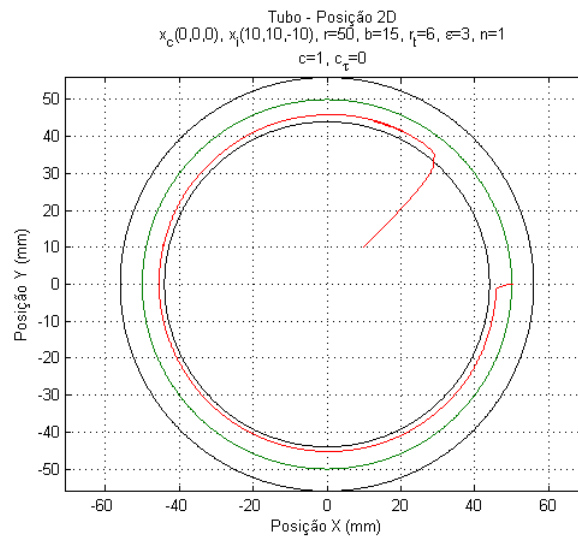


Figura 4.20 - Simulação do comportamento do elemento terminal para uma restrição do tipo Tubo, observada segundo a direcção Z.

De forma clara é ainda possível observar o desvio do elemento terminal face à curva que define o centro do tubo que está apresentado na Figura 4.21 .

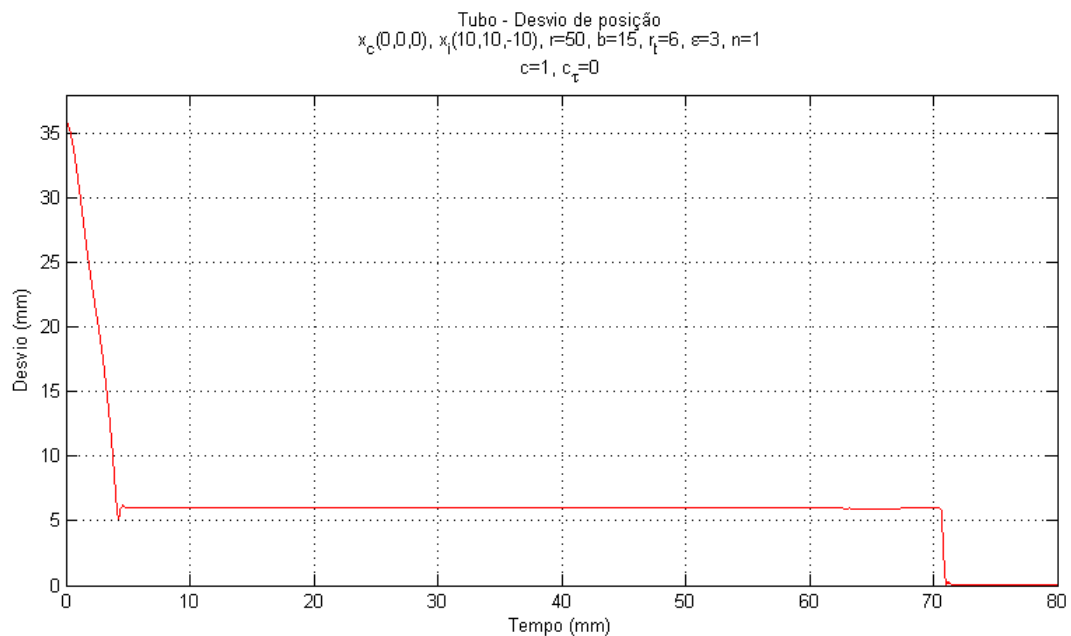


Figura 4.21 - Desvio de posição em relação a curva que define o centro da restrição virtual para uma restrição do tipo Tubo.

É possível observar que o elemento terminal entra no tubo e logo a seguir ultrapassa-o, fruto do ligeiro atraso existente nos controladores de velocidade. Depois desse momento o elemento terminal segue sempre no interior do tubo até encontrar o seu extremo, onde se imobiliza.

Foi ainda construído um outro bloco, Figura 4.22, para a restrição virtual Tubo com vista a estudar os parâmetros ϵ e n . Este novo bloco impõe uma restrição linear entre dois quaisquer pontos definidos pelo operador, \mathbf{x}_A e \mathbf{x}_B , utilizando para isso uma discretização da trajectória constituída por 10.000 pontos e as respectivas direcções preferenciais. O restante algoritmo da restrição virtual é em tudo semelhante ao caso da curva helicoidal.

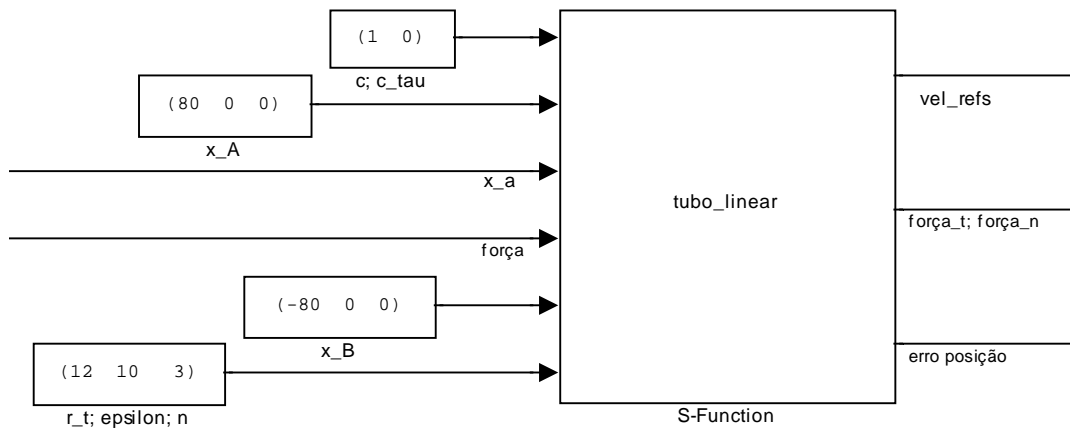


Figura 4.22 - Bloco Simulink que contém a restrição do tipo Tubo Linear.

Efectuaram-se simulações variando o parâmetro ϵ que define a região de transição dentro do tubo.

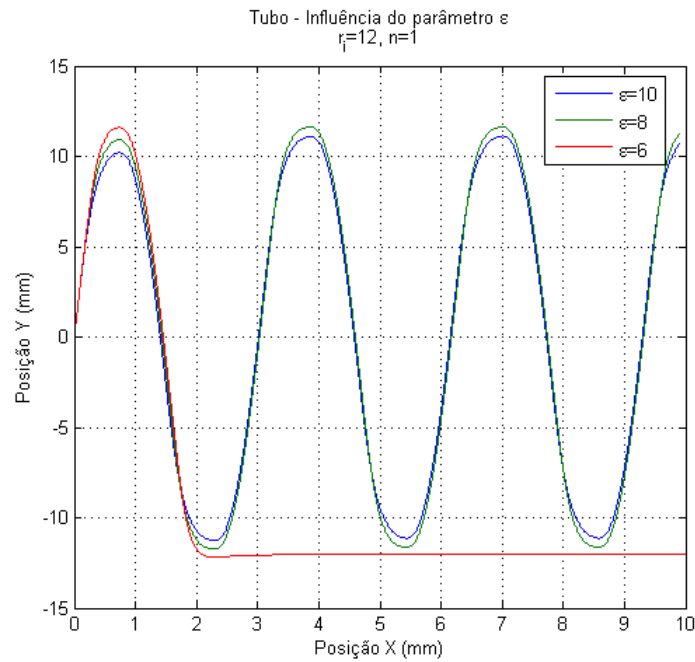


Figura 4.23 - Influência do parâmetro ϵ na restrição virtual Tubo.

Observa-se que quanto maior for a região de transição, menor valor de ϵ , mais difícil é ao operador manter o elemento terminal junto à curva que define o tubo. O elemento terminal tenderá a seguir junto à parede do tubo, para as mesmas condições de simulação. De notar que por vezes o elemento terminal ultrapassa a parede do tubo, esse efeito deve-se mais uma vez ao atraso dos controladores de eixo.

Na Figura 4.24 encontram-se ilustradas as simulações para diferentes valores do parâmetro n , o grau da função que define c_{tu} dentro da região de transição (4.18).

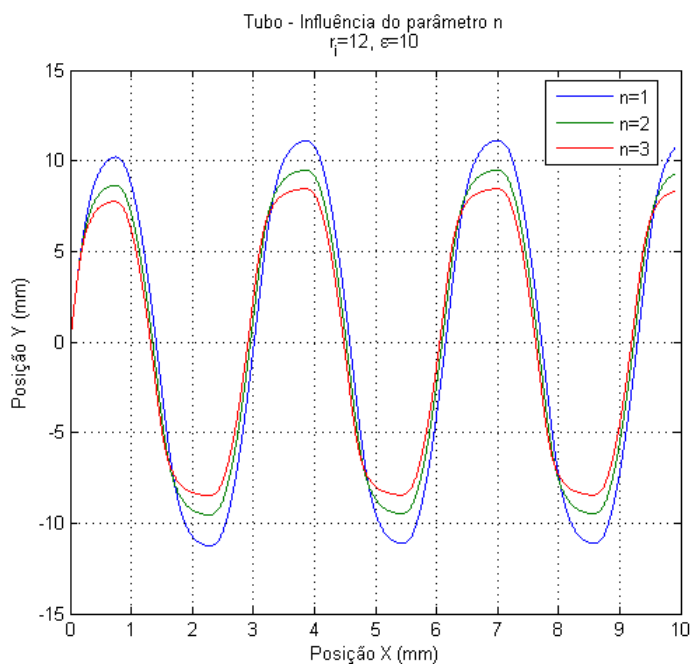


Figura 4.24 - Influência do parâmetro n na restrição virtual Tubo.

É possível observar que para a mesma região de transição, o aumento do parâmetro n impõe um movimento menos abrupto, ou seja mais preso. Isto é clarividente observando novamente a definição de c_{tu} na região de transição (4.18). À medida que n aumenta c_{tu} , no mesmo ponto no espaço, caminha para o c_t definido fora do tubo.

4.5.4 Cone

Nesta restrição virtual o utilizador é assistido de forma a movimentar o elemento terminal livremente dentro de um cone virtual, proporcionando assim uma sensação de funil virtual. Se o elemento terminal se encontrar fora do cone o utilizador é assistido de forma a dirigir-se para dentro deste.

O utilizador pode definir a posição do vértice do cone \mathbf{x}_v , a posição do centro da base \mathbf{x}_b e o ângulo α , em radianos, entre o eixo e a geratriz do cone. É ainda possível definir a região de transição ϵ_0 e o parâmetro que modela a superfície de transição n . O bloco *Simulink* construído para implementar a *S-Function* relativa à restrição virtual do tipo Cone está representado na Figura 4.25 .

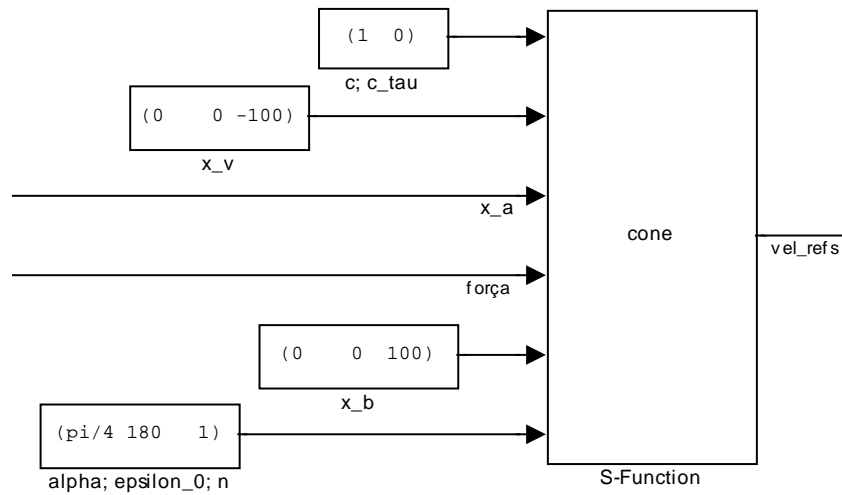


Figura 4.25 - Bloco do Simulink para a restrição virtual do tipo Cone.

Como não existe à partida uma trajetória definida deixa de ser necessária a rotina *mdlStart*. Na rotina *mdlOutputs* é aplicada a lei de controlo por admitância variável com o comportamento já descrito. A superfície de comutação para a lei de controlo proporcional pode ser observada na Figura 4.5 .

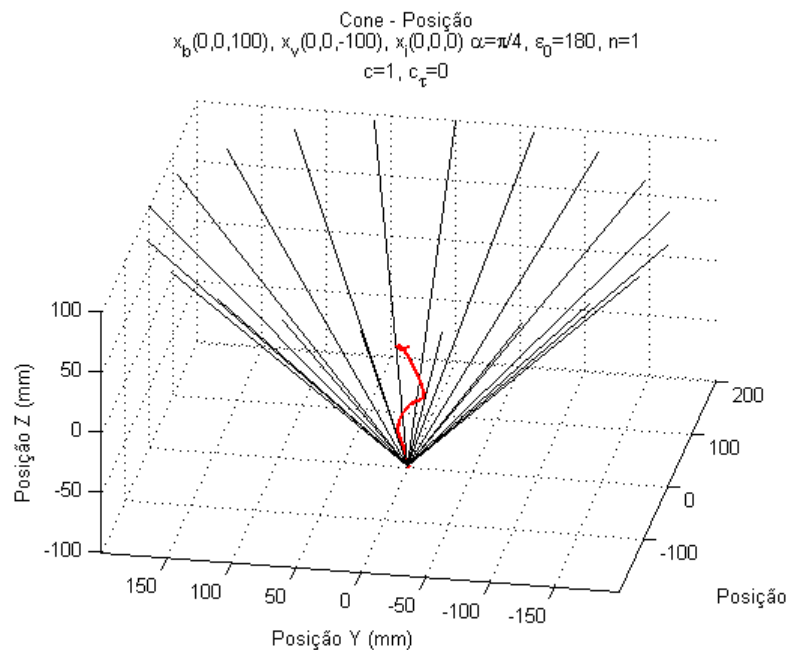


Figura 4.26 - Simulação do comportamento do elemento terminal para uma restrição virtual do tipo Cone.

É possível observar a vermelho a trajectória do elemento terminal durante a simulação. Neste caso o elemento terminal parte do interior do cone e apresenta um movimento livre até entrar dentro da zona de transição. Aí, devido aos parâmetros escolhidos, o elemento terminal “escorrega” pela face do cone até entrar na zona de controlo proporcional.

4.6 Conclusões

Neste capítulo foi efectuada uma descrição da estratégia de controlo por admitância para a aplicação de restrições virtuais. Assim foi descrita a formulação da lei geral de controlo por admitância variável no tempo. Para tal foi importante o conceito de resposta isotrópica/anisotrópica do sistema.

Para a implementação no espaço cartesiano foi assim preciso definir a direcção de referência δ e o erro cartesiano e como sendo a diferença entre o ponto da restrição e o ponto do elemento terminal num dado instante.

O comportamento do sistema perto do ponto alvo ou extremos de restrições Curva, Tubo e Cone é de suma importância e por isso foram descritas duas abordagens a esse problema.

Na implementação das restrições virtuais do tipo volumétricas utilizou-se a lei de controlo geral por admitância sendo a admitância e o erro funções definidas por troços.

Capítulo 5

Análise experimental das restrições virtuais

5.1 Introdução

Neste capítulo é efectuada uma análise aos dados recolhidos durante a fase de testes de implementação das diversas restrições. Os dados recolhidos variam com a restrição virtual testada pelo que nem todos os tipos de dados existem para as diversas restrições virtuais.

Esta análise surge como capítulo destacado, por mostrar o resultado da implementação do binómio controlador I-P/restrições virtuais. As restrições virtuais surgem assim associadas a um determinado controlador. Para outro controlador de admitância/velocidade os resultados seriam certamente diferentes.

Nos testes efectuados foi sempre imposto $c_\tau = 0$. Esta decisão deveu-se ao facto de ser essa a condição mais restritiva. Por outro lado para $c_\tau \neq 0$ tornava-se difícil concretizar os testes, visto que o operador não tinha uma noção clara do movimento que estava a realizar.

Os dados aqui representados foram gravados no *Target PC* durante a execução da aplicação de controlo e posteriormente transferidos para o *Host PC* para serem processados e editados.

Os dados foram recolhidos através dos dispositivos de medida existentes no sistema robótico, e que fornecem também dados para os diversos controladores e para a *Máquina de estados* (ex: *encoders*, transdutor de força).

Foram ainda efectuados testes de coordenação visual/motora para aferir qualitativamente o desempenho do sistema.

5.2 Ponto Alvo

Nestes testes o operador pode mover o elemento terminal segundo a direcção definida pela sua posição inicial \mathbf{x}_i e pelo ponto alvo \mathbf{x}_t .

Para testar a restrição virtual do tipo Ponto Alvo foram utilizados dois métodos. No primeiro o elemento terminal inicia o seu movimento no ponto $\mathbf{x}_i(0,0,0)$ tendo como ponto alvo o ponto $\mathbf{x}_t(150,100,0)$ (Figura 5.1). Apesar de o algoritmo de controlo contemplar as três direcções cartesianas, os movimentos na direcção Z foram desprezados. Considera-se que se trata de um movimento unicamente no plano XY. Este método permite ao operador ter uma melhor noção do ponto que se pretende atingir, servindo de base ao segundo método (se a restrição funcionasse no plano certamente iria funcionar no espaço cartesiano).

Na Figura 5.1 é possível observar que o movimento se efectua segundo a direcção desejada, nos dois sentidos.

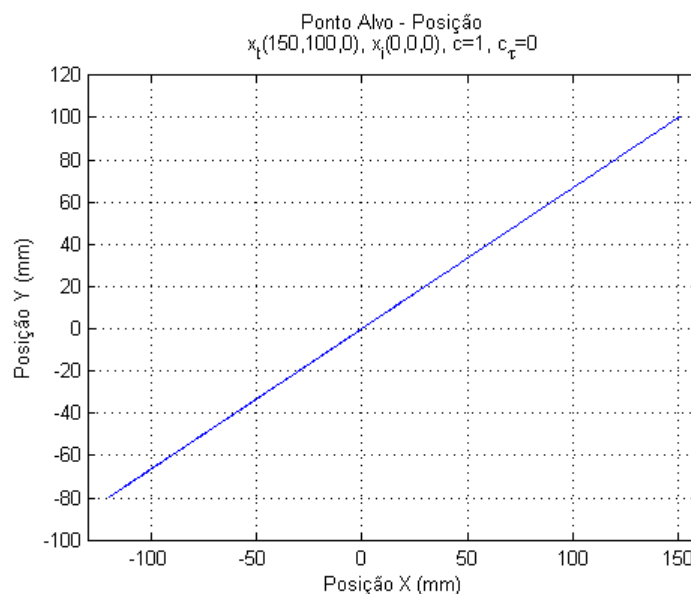


Figura 5.1 - Posição do elemento terminal para uma restrição do tipo Ponto Alvo, movimentação no plano XY.

É possível verificar que o elemento terminal percorreu um segmento recta com extremo em $\mathbf{x}_t(150,100,0)$. O operador foi capaz de movimentar o elemento terminal sem esforço em ambas as direcções, chegando mesmo a atingir o ponto $(-120,-80,0)$. O operador não foi capaz de movimentar o elemento terminal em qualquer outra direcção que não a direcção definida pelos pontos \mathbf{x}_i e \mathbf{x}_t . Quando se aproximou do ponto alvo o elemento terminal parou imediatamente devido à lei de controlo proporcional implementada. O operador não foi depois capaz de retirar o elemento terminal do ponto alvo.

Nos sucessivos movimentos de aproximação e afastamento ao ponto alvo o sistema apresentou pequeníssimos desvios à direcção inicial. Tais desvios não afectaram o desempenho geral da restrição virtual, visto que a direcção preferida é calculada tendo em conta a posição actual do elemento terminal \mathbf{x}_a . Os desvios podem ser explicados pelo atraso da resposta e pelo *overshoot* dos controladores.

No segundo método o elemento terminal inicia o seu movimento no ponto $\mathbf{x}_i(0,0,0)$ tendo como ponto alvo o ponto $\mathbf{x}_t(150,100,-50)$ (Figura 5.2). Este método já contempla assim um movimento nas três direcções do espaço cartesiano.

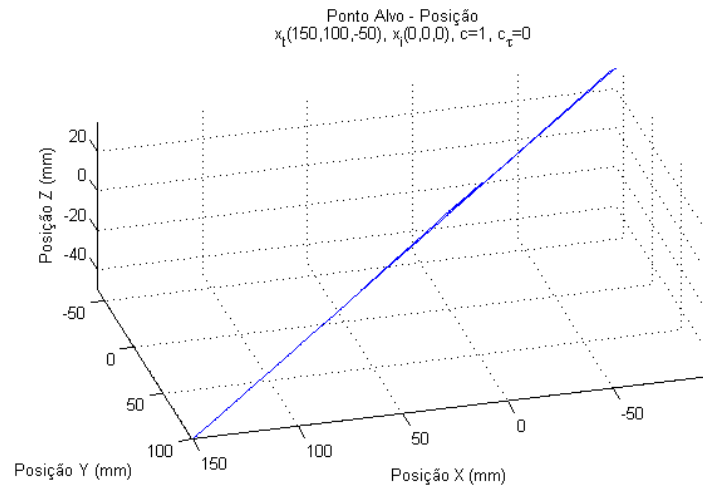


Figura 5.2 - Posição do elemento terminal para uma restrição do tipo Ponto Alvo, movimentação no espaço cartesiano.

Na Figura 5.2 observa-se que o sistema apresenta uma resposta semelhante à verificada para um movimento no plano XY (Figura 5.1). Também neste caso se verificaram pequenos desvios à direcção inicial, sendo que estes não comprometeram o desempenho geral da restrição virtual.

5.3 Curva

Neste teste pretende-se que o operador mova o elemento terminal sobre uma curva helicoidal definida no espaço pelo seu centro \mathbf{x}_c , raio r , e o parâmetro do passo b .

A restrição virtual do tipo Curva foi testada tendo como ponto de início de movimento o ponto $\mathbf{x}_i(0,0,0)$. O elemento terminal seguiu depois até ao ponto mais próximo da curva de acordo com a orientação da força exercida quando se encontrava no ponto inicial. O percurso do elemento terminal pode ser visualizado na Figura 5.3 .

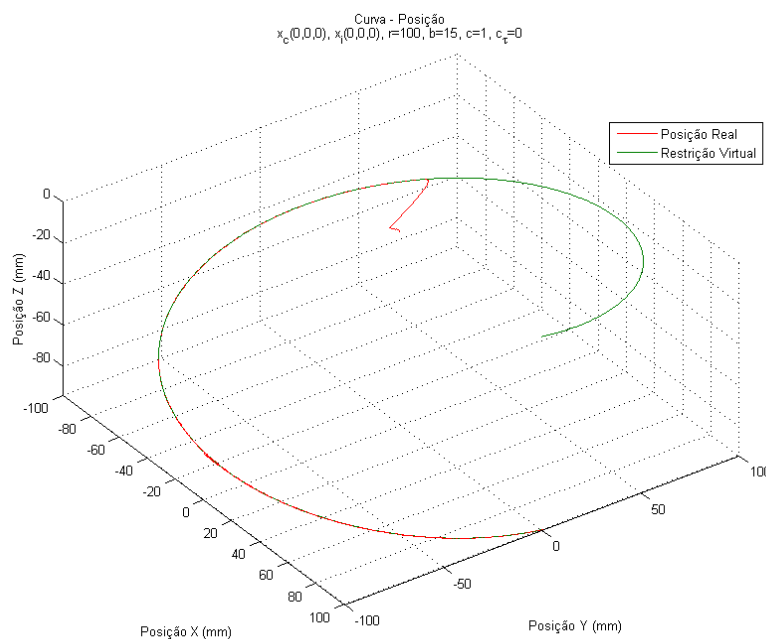


Figura 5.3 - Posição do elemento terminal para uma restrição do tipo Curva.

É possível observar que o elemento terminal se desloca sobre a restrição virtual até ao extremo da mesma. No extremo da restrição virtual o elemento terminal imobiliza-se devido à lei de controlo proporcional implementada.

Ao observar o erro de posição, apresentado na Figura 5.4, em relação à restrição virtual, desde o momento anterior a atingir a restrição virtual até o elemento terminal ficar imobilizado no final da mesma, verifica-se que o elemento terminal nunca se afasta mais do que 0.6 mm o que se pode considerar como um resultado satisfatório dada a natureza do sistema e os controladores implementados.

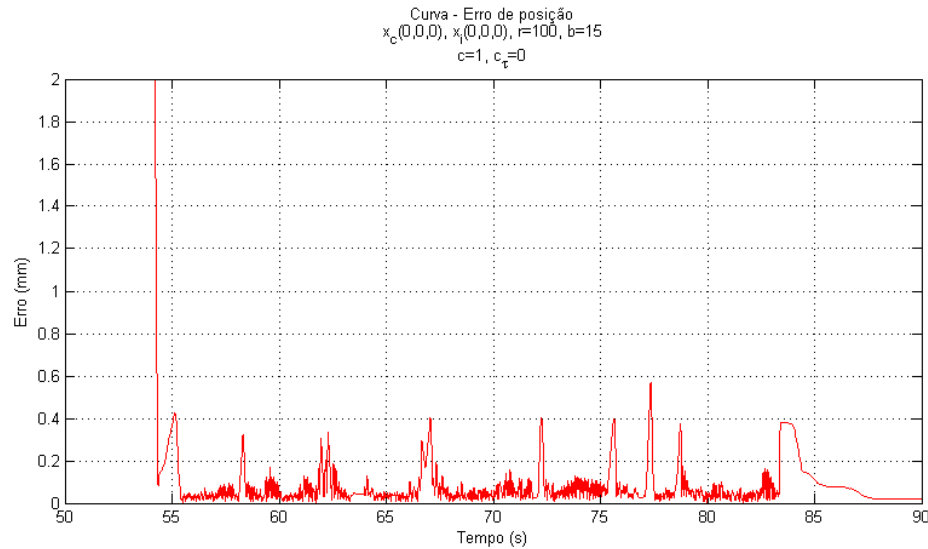


Figura 5.4 - Erro de posição em relação à curva.

É possível observar que nos instantes iniciais o elemento terminal aproxima-se da restrição virtual, cruza-a uma primeira vez e só instantes depois volta a estar sobre ela. Este efeito não se verifica na simulação (Figura 4.15), onde o elemento terminal faz o seguimento da curva logo a partir do instante em que cruza a restrição virtual. Este feito de sobre-elongação tem por causa a resposta do controlador que não é capaz de reduzir a velocidade num curto espaço de tempo.

No final da trajectória, aos 83s, é possível observar que o erro diminui de forma suave até ser nulo. Esta diminuição é fruto do controlo proporcional aplicado na extremidade da restrição virtual.

Devido à inexistência de uma relação visual/motora o operador teve alguma dificuldade em se aperceber que a trajectória que estava a executar era uma curva helicoidal.

5.4 Tubo

Nestes testes pretende-se que o operador mova o elemento terminal para o interior de um volume. Uma vez no seu interior o operador deve ser capaz de mover o elemento terminal

igualmente em todas as direcções, bem como deve ser impedido de sair do volume previamente definido.

A restrição virtual do tipo Tubo foi testada tendo como ponto de partida do elemento terminal o ponto $x_i(0,0,0)$. O tubo, em formato helicoidal desenvolve-se no sentido negativo do eixo Z. A zona de transição do tubo tem 2 mm de espessura ($r - \epsilon$) e c_{tu} é calculado nessa zona por uma função de terceira ordem (4.18). Estes parâmetros foram escolhidos tendo em conta as simulações realizadas.

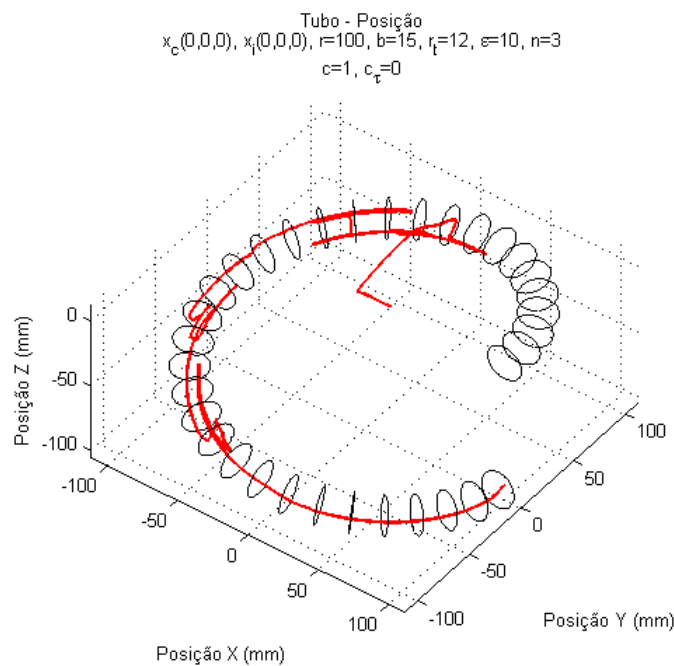


Figura 5.5 - Posição do elemento terminal para uma restrição virtual do tipo Tubo.

É possível observar a vermelho na Figura 5.5 que, primeiramente o elemento terminal dirige-se em direcção ao tubo e, dentro deste o movimento se torna livre. Uma vez no interior do tubo o operador é impedido de ultrapassar a superfície virtual. Quando o operador se aproximou de um dos extremos do tubo, o elemento terminal convergiu, para o ponto central da secção extrema do tubo.

Foram ainda recolhidos dados relativos ao desvio de posição que o elemento terminal apresentou relativamente à curva que define o centro do tubo. Tais dados encontram-se apresentados na Figura 5.6 .

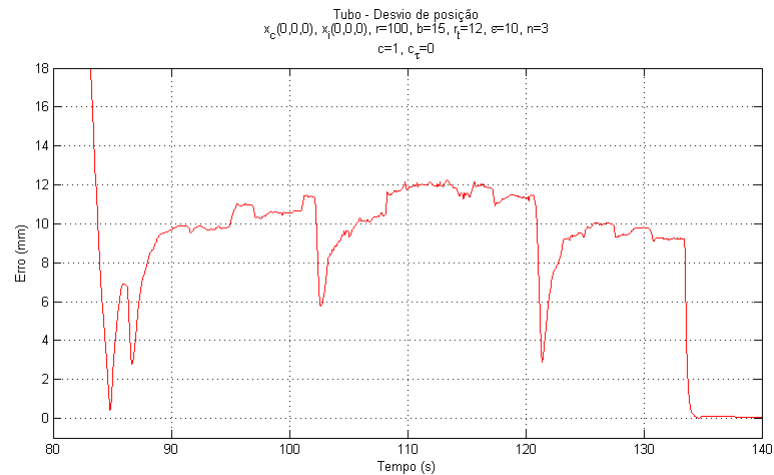


Figura 5.6 - Desvio de posição em relação ao centro da secção do tubo.

Pode afirmar-se que uma vez dentro do tubo o elemento terminal terá ultrapassado a superfície virtual, num máximo de 0.3 mm mas por breves instantes. Tendo em conta a resposta dos controladores considerou-se este resultado aceitável.

Como se tratava de um tubo com uma forma helicoidal foi difícil ao operador distinguir a secção de movimento livre pelo que se tornou mais fácil fazer “escorregar” o elemento terminal pela parede do tubo. É fácil notar a zona de transição, faixa 10-12 mm, bem como a dificuldade de movimentar o elemento terminal dentro da mesma.

Foi ainda testada uma restrição virtual do tipo Tubo com uma trajectória linear. Os resultados, apresentados na Figura 5.7, estão de acordo com os anteriores e dentro do esperado.

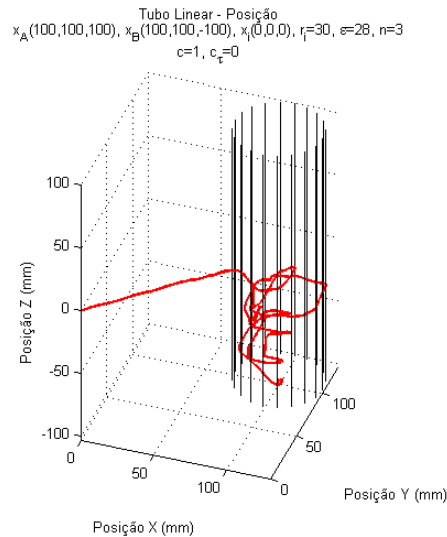


Figura 5.7 - Posição do elemento terminal para uma restrição virtual do tipo Tubo Linear.

Esta restrição virtual, com um tubo em formato cilíndrico, permitiu ao operador ter uma melhor noção da superfície virtual e desse modo tornou mais agradável a movimentação no seu interior. De notar que a orientação escolhida para este teste ajudou a que o operador não estivesse sempre “colado” à superfície virtual.

5.5 Cone

Nestes testes pretende-se que operador mova o elemento terminal para o interior de um cone virtual. Uma vez no seu interior o operador deve ser capaz de mover o elemento terminal igualmente em todas as direcções. Pretende-se obter o efeito de funil virtual, o que levará o operador a mover intuitivamente o elemento terminal até ao vértice do cone.

Na restrição virtual do tipo Cone foram testadas duas situações. Na primeira situação o elemento terminal inicia o seu movimento já dentro do cone enquanto que na segunda situação o elemento terminal encontra-se fora do cone.

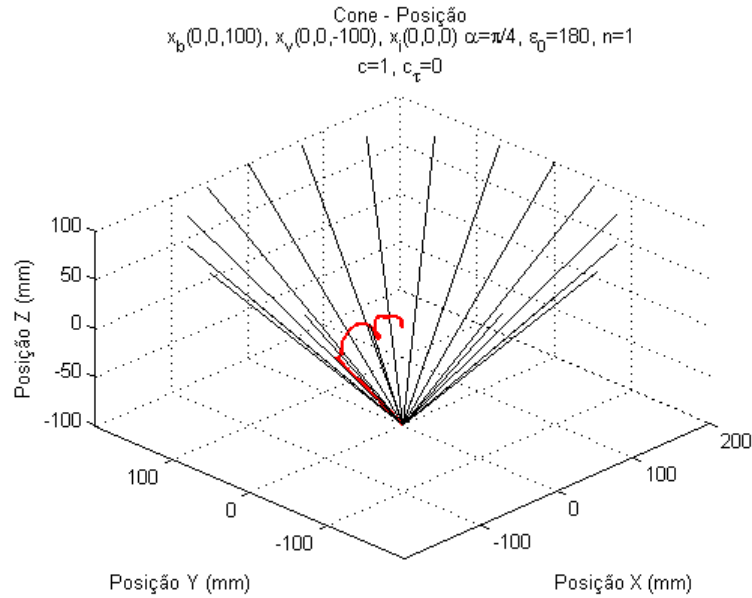


Figura 5.8 - Posição do elemento terminal para uma restrição virtual do tipo Cone em que o movimento se inicia dentro.

Na Figura 5.8 é possível observar o movimento do elemento terminal dentro de um cone. Verificou-se que existia um movimento livre dentro do cone e fora da zona de transição. Quando o elemento terminal entrou na zona de transição não foi possível voltar à zona completamente livre, apenas foi possível deslocá-lo em direcção ao vértice do cone. Isto deveu-se ao facto de a zona de transição ser demasiado grande. Assim junto à fronteira do cone c_{co} tomava valores próximos de zero (4.30), o que tornava impossível o movimento em outra direcção que não em direcção ao vértice.

No segundo teste, Figura 5.9 foi aplicada uma zona de transição mais pequena, 2 mm, e o c_{co} dentro da zona de transição passou a ser calculado por uma função de terceira ordem (4.30).

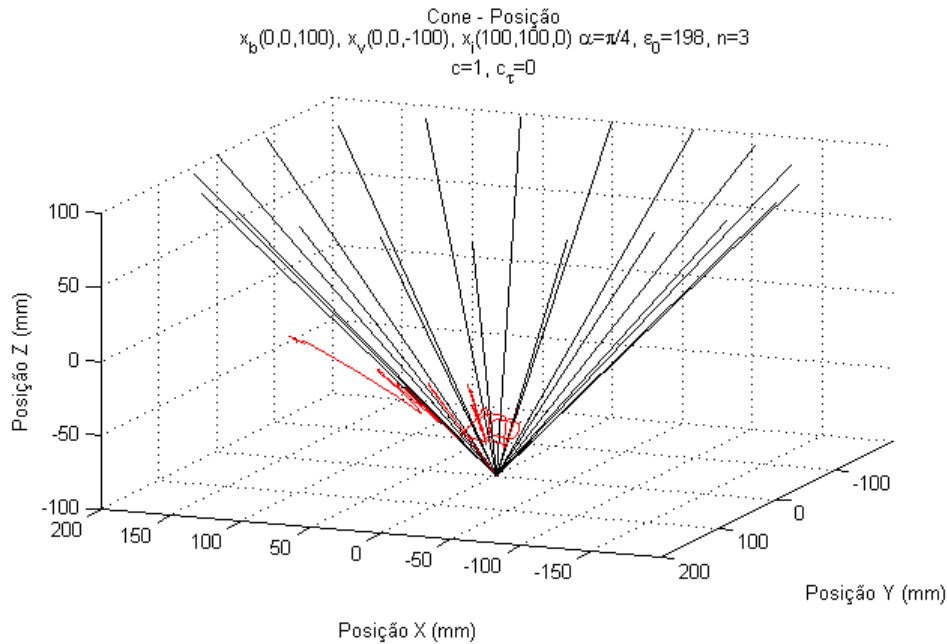


Figura 5.9 - Posição do elemento terminal para uma restrição virtual do tipo Cone em que o movimento se inicia fora.

Neste teste é possível observar que o elemento terminal se aproxima da superfície do cone, preferencialmente numa direcção directa ao vértice. Foi necessário um pouco mais de esforço para mover o elemento terminal em direcção ao cone. Uma vez dentro do cone o elemento terminal apresentou uma resposta livre. Ao aproximar-se do vértice o elemento terminal deslocou-se para este segundo a lei de controlo proporcional.

5.6 Análise da coordenação visual/motora

No decorrer dos testes foi perceptível a dificuldade do operador em compreender o ambiente virtual criado. O operador não conseguia, por exemplo, saber em que ponto da curva se encontrava ou se, no caso do Tubo e Cone, estava junto à superfície. Esta falta de elementos referenciais obrigava a um esforço maior do operador no seguimento de uma curva uma vez que, ao não saber onde se encontrava, era obrigado a forçar o movimento em várias direcções até encontrar uma direcção preferencial de movimento.

Para aferir melhor a resposta do operador ao ambiente com e sem restrições virtuais foi pedido ao operador que tentasse desenhar uma circunferência sobre uma folha de papel, primeiramente utilizando uma restrição virtual e seguidamente repetir o mesmo procedimento mas com um ambiente completamente livre (Figuras 5.10, 5.11 e 5.12).

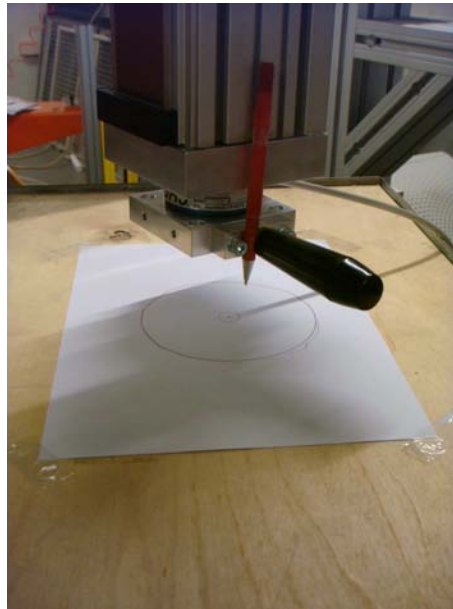


Figura 5.10 - Teste de coordenação visual/motora.

No movimento com restrição virtual, Figura 5.11, o operador conseguiu cumprir a trajetória com rapidez e eficácia, não sendo necessário esforço físico. O operador foi auxiliado pelo robô. O operador tinha a noção de qual a trajetória que tinha seguido e em que ponto do espaço de trabalho se encontrava num determinado momento.

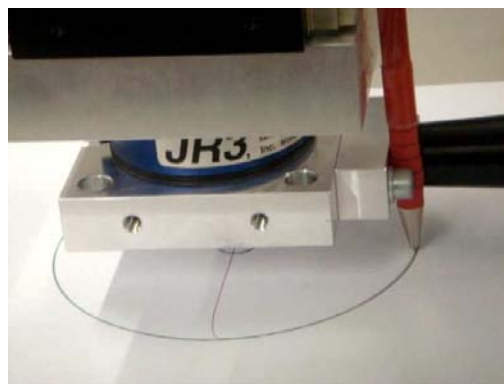


Figura 5.11 - Execução de uma restrição virtual do tipo Curva.

No teste seguinte, Figura 5.12, foi pedido que num ambiente sem restrições virtuais o operador tentasse reproduzir uma curva previamente traçada. O operador conseguiu esboçar a curva mas não a conseguiu cumprir na íntegra. O esforço físico e mental para completar tal tarefa foi mais elevado uma vez que exigiu uma coordenação das diversas articulações da mão, braço e antebraço, juntamente com a resposta do manipulador.

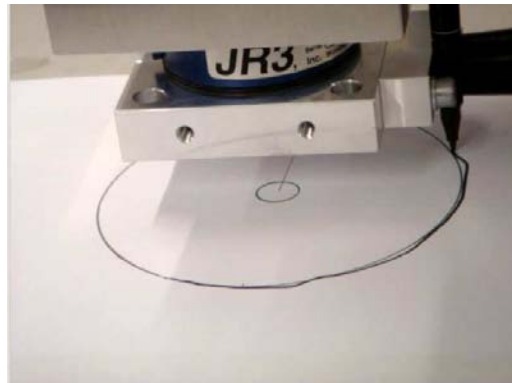


Figura 5.12 - Tentativa de seguimento de um contorno sem restrição virtual.

5.7 Conclusões

Neste capítulo foi efectuada uma análise quantitativa e qualitativa da solução de controlo recorrendo a restrições virtuais. Para cada tipo de restrição virtual foram identificadas as condições de teste bem como feita uma descrição do comportamento do sistema robótico durante cada teste.

Na análise dos resultados obtidos foi feita a ponte com as situações simuladas com o intuito de identificar semelhanças e descobrir as razões de comportamentos díspares.

Caso não existam referências no espaço de trabalho, os movimentos do elemento terminal são de difícil percepção para o operador. A resposta do operador melhora qualitativamente se este tiver uma noção clara do movimento ou do objectivo a atingir.

O sistema testado deve ser visto como um todo, sendo o comportamento do controlo com restrições virtuais indissociável dos controladores de eixo. Concluiu-se que uma boa escolha

de alguns parâmetros (ϵ e n) permite compensar os defeitos do controlador de velocidade/admitância I-P.

Concluiu-se que o sistema cumpre de uma forma geral as restrições ainda que condicionado pela resposta do controlador de velocidade I-P.

Capítulo 6

Conclusões e desenvolvimentos futuros

6.1 Conclusões

Este trabalho foi desenvolvido com o intuito de criar um protótipo de desenvolvimento e aprendizagem na área de controlo de sistemas robóticos, mais concretamente na área dos sistemas colaborativos.

Na pesquisa de informação para este trabalho foi possível observar que os sistemas colaborativos estão numa fase de forte implementação nomeadamente nas áreas médico-cirúrgica, reabilitação e industrial-automóvel. Apesar disso há ainda um vasto conjunto de áreas onde esta tecnologia pode ser aplicada, com destaque para a programação de pontos/tarefas em robôs industriais.

No decorrer do trabalho foi criada uma aplicação de controlo (*Matlab/Simulink/xPC Target*) que possibilitou a operação do robô cartesiano através da interacção directa de um operador humano. A aplicação de controlo é composta por dois níveis. O controlo comportamental que determina a resposta do sistema de acordo com o valor das receptividades num dado momento e apresenta uma lógica sequencial de acordo com o diagrama de estados do sistema. O segundo nível de controlo determina o comportamento do elemento terminal quando um dado estado se encontra activo. Neste segundo nível de controlo foram desenvolvidos algoritmos de controlo que permitem o controlo de velocidade do robô cartesiano durante o procedimento de posicionamento no ponto de referência do espaço de trabalho (*Zero da Máquina*), algoritmos de controlo que permitem o controlo de

posição do robô em diferentes pontos do espaço de trabalho, e ainda algoritmos de controlo que permitem uma manipulação colaborativa do elemento terminal.

Para que o sistema pudesse ser manipulado por um operador humano foi necessário criar uma interface física. Esta interface permite um controlo do elemento terminal confortável graças ao formato ergonómico do punho criado. A interface física cumpre ainda funções de segurança, com a inclusão de um botão de homem-morto, e funções de comando, com a inclusão de um botão para activação de uma ferramenta pneumática.

A utilização do módulo de controlo *xPC Target (Matlab)* permitiu o desenvolvimento de uma arquitectura de controlo de tempo-real que se verificou eficaz na implementação dos diversos algoritmos de controlo. A arquitectura *Host/Target PC* permite o fácil desenvolvimento e simulação dos algoritmos de controlo num *PC (Host PC)*. Depois de desenvolvidos os algoritmos a aplicação de controlo é compilada e transferida para o *Target PC*, onde é executada em tempo-real.

A implementação da manipulação colaborativa simples (sem restrições virtuais) ditou o tipo de controlo: admitância em detrimento do controlo por impedância. Foi descrita a estratégia de controlo por admitância para a aplicação das restrições virtuais, sendo para isso formulada a lei geral de controlo por admitância variável no tempo. Foram explicados os algoritmos de controlo para cada tipo de restrição virtual adoptada. Esses algoritmos foram depois transpostos para ficheiros executáveis *.MEX* utilizados pelas *S-Function* do *Matlab/Simulink*.

Nesta dissertação foram apresentados e comentados alguns resultados experimentais decorrentes da manipulação do robô cartesiano. Esses resultados demonstram o comportamento dos controladores de velocidade e a implementação da manipulação colaborativa utilizando restrições virtuais, esta última baseada num controlo por admitância. Tratando-se de uma interacção Homem-Máquina procedeu-se à análise da percepção que o operador tem sobre do comportamento da máquina, através da realização de diversos ensaios experimentais.

De momento o sistema robótico de manipulação colaborativa utilizando restrições virtuais desenvolvido apresenta um controlo que permite ao operador controlar de forma segura, confortável, sem esforço e intuitiva, o elemento terminal em vários ambientes virtuais.

6.2 Desenvolvimentos futuros

O trabalho desenvolvido teve como objectivo a criação de um protótipo para a implementação e teste do controlo por admitância utilizando restrições virtuais. Neste sentido são várias as melhorias e desenvolvimentos que podem valorizar o trabalho efectuado até ao momento, das quais se destacam as seguintes:

- Desenvolvimento de controladores de eixo robustos, nomeadamente controladores de estrutura variável que permitam melhorar a resposta do sistema ao controlo por admitância.
- Criação de uma interface gráfica amigável (*user-friendly*) em plataformas *Microsoft Visual Basic/Visual Studio C#*. Essa interface gráfica teria como objectivos o controlo do sistema robótico e a possibilidade de o operador, fácil e rapidamente, poder definir o tipo de restrição virtual e os seus parâmetros.
- Melhoria da sensorização do sistema robótico. Nomeadamente a destacar o uso de um transdutor de força mais imune a interferências eléctricas.
- Estudo do comportamento dinâmico da estrutura do sistema robótico. Neste trabalho foram considerados os eixos como sendo elementos independentes/desacoplados, com inércias fixas. Na realidade tal não acontece. Há ainda que contar com a variação da rigidez imposta pela interacção do elemento terminal com o operador.

Referências

- Almeida, F. G. (2009). *Apontamentos da disciplina de Sistemas de Controlo*. Porto: Faculdade de Engenharia da Universidade do Porto.
- Aström, K. J. (1997). *Computer-controlled systems : theory and design*. 3rd ed. Upper Saddle River, NJ: Prentice-Hall.
- Bettini, A., Marayong, P., Lang, S., & Okamura, A. (2004). Vision-Assisted control for Manipulation Using Virtual Fixtures. *IEEE Transactions on Robotics*, Vol. 20, N°6 , pp 953-966.
- Book, W., Charles, R., Davis, H., & Gomes, M. (1996). The Concept and Implementation of a Passive Trajectory Enhancing Robot. *1996 International Mechanical Engineering Congress and Exposition (IMECE)*, (pp. 633-638). Atlanta, GA,.
- Cardoso, M. J. (2009). *Concepção e Implementação de uma Estratégia de Controlo Aplicada a um Sistema Robótico do tipo Master/Slave*. Tese de Mestrado. Faculdade de Engenharia da Universidade do Porto. Portugal.
- Colgate, J. E., Peshkin, M., & Klostermeyer, S. H. (2003). Intelligent Assist Devices in Industrial Applications: A Review. *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, (pp. 2516-2521). Las Vegas, Nevada.
- Davies, B., Harris, S., Jakopc, M., Gomes, P., & Rodriguez-y-Baena, F. (26 de Abril de 2004). *Country Doctor*. Obtido em 3 de Março de 2010, de Orthopaedic Robotics: <http://www.countrydoctor.co.uk/education/Education%20-%20Orthopaedic%20robotics.htm>
- Emeagwali, I. (2004). Performance Analysis of Steady-Hand Teleoperation versus Cooperative Manipulation. *2004 Symposium on Hapt Interfaces for Virtual Environment and Teleoperator Systems, IEEE Virtual Reality*. Chicago, IL.
- Feio, J. E. (2008). *Controlo de Impedância de Robôs Manipuladores para Aplicações em Cirurgia Ortopédica*. Tese de Mestrado. Instituto Superior Técnico, Portugal.
- Hocoma. (s.d.). *Hocoma, We move you*. Obtido em 3 de Maio de 2010, de Lokomat® - Features & Functions: <http://www.hocoma.com/en/products/lokomat/features-functions/>
- Hogan, N. (1985). Impedance control: an approach to manipulation: Part I-III. *ASME Journal of Dynamic Systems, Measurement and Control* , Vol. 107, pp 1-24.
- Intelligent Machine Dynamics Laboratory, G. T. (s.d.). *Intelligent Machine Dynamics Laboratory*. Obtido de Haptic Robotics: http://www.imdl.gatech.edu/haptics/index_main.html
- Kazerooni, H. G. (1993). Human Extenders. *ASME Journal of Dynamic Systems, Measurement and Control* , Vol. 115, pp 281-290.
- Laboratory for Intelligent Mechanical Systems, N. U. (19 de Novembro de 2004). *The Haptic Community Web Site*. Obtido em 8 de Março de 2010, de Haptics Photo Gallery: <http://haptic.mech.northwestern.edu/intro/gallery/padyc.jpg>

- Lammertse, P. (2004). Obtido em 22 de Fevereiro de 2010, de http://www.haptist.com/resources/Lammertse_2004_admittance_impedance_control_dual.pdf
- Marayong, P. (2007). *Motion Control Methods for Human-Machine Cooperative Systems*. PhD Thesis. John Hopkins University, USA.
- Moore Jr., C. A. (2001). *Design, Construction, and Control of a 3-Revolute Arm Cobot*. Evanston, Illinois: PhD Thesis. Northwestern University.
- Ogata, K. (1997). *Modern Control Engineering 3rd ed*. New Jersey: Prentice Hall Internacional.
- Okamura, A. M. (2000). *Haptic Exploration of Unknown Objects*. Stanford, CA, USA: PhD Thesis. Stanford University.
- Peshkin, M., & Colgate, J. E. (1999). Cobots. *Industrial Robot: An Internacional Journal* , Vol. 26, Nº 5, pp 335-341.
- Peshkin, M., Colgate, J. E., Wannasuphoprasit, W., Moore, C. A., Gillespie, R. B., & Akella, P. (2001). Cobot Architecture. *IEEE Transactions on Robotic and Automation* , Vol. 17, Nº 4, pp 337-390.
- Rosenberg, L. B. (1993). Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation. *Proc. IEEE Virtual Reality Int. Symp. (VRAIS'93)* , pp 76-82.
- Schneider, O. T. (2000). PADyC: a Synergistic Robot for Cardiac Puncturing. *Proceedings of the 2000 IEEE Internacional Conference on Robotics & Automation* (pp. 2883-2888). São Francisco: IEEE.
- Szondy, D. H. (s.d.). Obtido em 9 de Março de 2010, de <http://davidszondy.com/future/robot/hardiman.htm>
- The Math Works. (2002). *Simulink model-based and system-based design : writing S-functions : Version 5*. Natick, MA: The MathWorks.
- Troccaz, J., & Delnondedieu, Y. (1996). Semi-active guiding systems in surgery. A two-dof prototype of the passive arm with dynamic constraints (PADyC). *Mechatronics* , Vol.6, Nº. 4, pp 399-421.
- Tung, J. (2002). *xPC DSP System*. Obtido em 25 de Março de 2010, de xPC DSP System: <http://www.bmed.mcgill.ca/reklab/ARCHIVE/xPC/xPC.htm>
- Wannasuphoprasit, W. (1999). *Cobots: Collaborative Robots*. Evanston, Illianois: PhD Thesis. Northwestern University.

